

RAZPOZNAVANJE 3-DIMENZIONALNIH OBJEKTOV NA POLJUBNEM OZADJU

Katarina Mele

MAGISTRSKA NALOGA

predložena

Fakulteti za računalništvo in informatiko

Univerze v Ljubljani

kot delna izpolnitev pogoja za pridobitev naslova

magistrica računalništva in informatike

Ljubljana, 2003

Mentorica:

doc. dr. Jasna Maver

Magistrska naloga je bila izdelana pod mentorstvom doc. dr. Jasne Maver in je last Fakultete za računalništvo in informatiko v Ljubljani. Za objavlanje in uporabo rezultatov magistrskega dela je potrebno soglasje zgoraj omenjene ustanove.

Besedilo je oblikovano s sistemom za pripravo besedil L^AT_EX.

Povzetek

Pri glediščno osnovanih metodah nam za učenje razpoznavanja 3-dimenzionalnih objektov služi zgolj vizualni izgled objektov na slikah. Vizualne značilnosti objektov zajamemo z dovolj veliko množico slik, ki prikazujejo različne izgleda istega objekta. To pomeni, da so slike objektov lahko zajete iz različnih zornih kotov, pod različnimi osvetlitvami, pri različnih razdaljah do objekta in v različnih prizorih.

V delu smo problem učenja razpoznavanja omejili na različni zorni kot in različna ozadja, na katerih se objekti nahajajo. Problem učenja razpoznavanja smo reševali s pomočjo metode podpornih vektorjev. Metoda podpornih vektorjev v splošnem ne ločuje med slikovnimi elementi, ki pripadajo objektu in slikovnimi elementi, ki pripadajo ozadju.

S spreminjanjem kota pogleda na objekt se spreminja njegova silhueta na sliki in posledično tudi delež in položaj slikovnih elementov, ki pripadajo objektu. Če spreminjamo tako zorni kot kot tudi ozadje, se učinkovitost klasifikatorja zelo poslabša. Pri nepoznanem številu objektov in njihovi lokaciji na sliki se zahtevnost problema še dodatno poveča.

Če imamo v fazi učenja pogoje, ki omogočajo segmentacijo objekta od ozadja, je smiselno, da učenje razpoznavanja usmerimo tako, da zmanjšamo vpliv slikovnih elementov ozadja. Možnih je več pristopov. V delu so predstavljeni trije. Razvili smo hierarhično SVM-metodo in jo primerjali z robustno PCA- in BW-metodo.

Prvo tehniko, metodo BW (ang. *Black-White*), uvrščamo med metode pedagoškega učenja. V fazi učenja točke ozadja nadomesti z ekstremnimi vrednostmi. Drugi dve tehniki, hierarhični SVM in robustni PCA, izvajata klasifikacijo le na podmnožicah slikovnih elementov na slikah in tako izločita tiste slikovne elemente, ki slabijo klasifikacijo. V primeru hierarhičnega SVM-ja organiziramo slikovne elemente na slikah, ki pripadajo

objektu, v drevesno strukturo. V skladu z drevesno strukturo zgradimo hierarhični klasifikator, ki pri učenju sloni le na slikovnih elementih objekta.

Metoda glavnih komponent (ang. *Principal Component Analysis*, PCA) omogoča v procesu razpoznavе tudi rekonstrukcijo objekta. Robustni PCA uporablja večje število hipotez, izračunanih na različnih podmnožicah na sliki. S pomočjo izračunanih rekonstrukcij najprej izloči slikovne elemente ozadja, nato pa izvede proces razpoznavе.

Vse tri metode smo implementirali in njihovo učinkovitost analizirali na podlagi empiričnih rezultatov. Rezultati kažejo, da naša hierarhična metoda zniža odstotek napačno pozitivnih in pri tem ohrani visoko stopnjo pravilno razpoznanih.

Ključne besede: glediščno osnovane metode, razpoznavanje objektov, poljubno ozadje, metoda podpornih vektorjev, robustni PCA, BW-metoda.

Abstract

Learning and recognition of 3-D objects with appearance-based methods rely solely on visual properties of the objects. Visual appearances of an object are acquired by a sufficient number of images showing the object under different conditions and in different situations including varying angles of poses, changing illumination, different distances of the object from the camera, and cluttered background.

In the present work appearance variations are conditioned only by different object poses and cluttered backgrounds. As a learning method the Support Vector Machine (SVM) is used. The classic SVM classifier does not distinguish the background from the object. By changing the view angle the shape of the object silhouette in an image changes. Consequently, the number and the locations of the objects and background vary from view to view. On condition that both, the pose of the object and the background vary, the efficiency of the standard classification methods drops significantly. Furthermore, if the positions and the number of the objects in the image are not given, the complexity of the problem increases.

If the training set allows object segmentation, it is reasonable to employ this knowledge in the process of recognition training and reduce the influence of the background as much as possible. Several approaches are possible. We developed Hierarchical SVM method and compared it with Black-White method (BW) and Robust Principal Component Analysis (PCA).

BW method is one of the pedagogical learning methods. In the learning phase the background pixels are replaced with extreme values. The hierarchical SVM and the robust PCA both rely on a subset of image pixels. Hierarchical SVM organizes the object pixels into hierarchical tree structure. In accordance with the hierarchical structure the

hierarchical classifier is based solely on object pixels. In addition, PCA enables the reconstruction of the object in the recognition phase. The method starts with several hypothesis using different subsets of image points. With respect to reconstruction errors the points belonging to the background are first rejected and then the recognition proceeds.

All three proposed methods are implemented and evaluated experimentally. The experimental results show that our Hierarchical SVM reduces false positive rate and maintains similar high recognition rate compared to the other two methods.

Key words: appearance-based recognition, object recognition, cluttered background, support vector machine, robust PCA, BW-method.

Kazalo

Kazalo slik	xvii
Kazalo tabel	xix
Kazalo algoritmov	xxi
1 Uvod	1
1.1 Pomen vizualne informacije	1
1.2 Problemi pri razpoznavanju objektov na podlagi vizualne informacije . . .	2
1.3 Metode računalniškega vida za razpoznavanje 3-D objektov	2
1.4 Cilji naloge	4
1.5 Pregled sorodnih del	4
1.6 Struktura naloge	7
2 Metoda podpornih vektorjev	9
2.1 Formulacija problema	9
2.2 Določanje vezanega ekstrema	11
2.3 Mehka meja	13
2.4 Uporaba jedra	16
2.5 Metoda podpornih vektorjev za klasifikacijo v več razredov	19
2.5.1 Eden proti vsem ostalim	19
2.5.2 Eden proti enemu	20
2.5.3 DAGSVM	20
2.5.4 Teniški turnir	21
2.5.5 k-razredni SVM	21

3	Metoda podpornih vektorjev enega razreda	23
3.1	OSH, ki ločuje koordinatno izhodišče od podatkov učne množice	24
3.2	Hiperkrogla	26
4	Metoda BW	29
4.1	Ideja BW-metode	29
4.1.1	Delež slikovnih elementov, ki pripadajo ozadju, je majhen	30
4.1.2	Delež slikovnih elementov, ki pripadajo ozadju, je relativno velik	31
4.2	Barvna BW-metoda	32
5	Metoda glavnih komponent	35
5.1	Osnovni PCA	35
5.2	Robustni PCA	37
5.2.1	Generiranje hipotez	37
5.2.2	Izbira hipoteze	38
6	Hierarhični SVM	41
6.1	Definicija osnovnih pojmov	41
6.1.1	Maska	41
6.1.2	Maska preseka	41
6.1.3	Maska unije	42
6.1.4	Velikost maske	42
6.2	Hierarhična ureditev	42
6.2.1	Delitev množice	43
6.3	Hierarhija SVM-jev enega razreda	48
6.3.1	Gradnja hierarhije	48
6.3.2	Faza razpoznavanja	48
6.4	Ločitev prekrivajočih se krogel z dvorazrednim SVM-jem	50
6.4.1	Faza gradnje dvorazrednih SVM-jev	51
7	Testi	55
7.1	Baza figuric	55
7.2	Testiranje barvne BW-metode	56

7.2.1	Učna faza	56
7.2.2	Testna faza in rezultati	57
7.2.3	Občutljivost na premik	58
7.3	Testiranje hierarhičnega pristopa	61
7.3.1	Učna faza	61
7.3.2	Testna faza in rezultati	61
7.4	Testiranje robustnega PCA pristopa	64
7.4.1	Učna faza	64
7.4.2	Testna faza in rezultati	64
7.4.3	Občutljivost na premik	70
7.5	Povzetek testiranj metod	74
7.6	Primerjava BW-, hierarhične SVM- in robustne PCA-metode	77
7.7	Baza COIL-100	82
7.8	Barvna BW-, hierarhična SVM- in robustna PCA-metoda na COIL-100 bazi	84
7.8.1	Barvna BW-metoda	85
7.8.2	Hierarhični SVM	88
7.8.3	Robustna PCA-metoda	90
8	Zaključek in nadaljnje delo	97
9	Dodatek A	101
9.1	Notacija	101
	Zahvala	103
	Izjava	109

Slike

2.1	Grafični prikaz OSH-ja za enostaven dvodimenzionalni primer	10
2.2	Primer SVM-klasifikatorja	13
2.3	OSH za linearno neločljivo množico	16
2.4	Ideja jedra	16
2.5	Primer uporabe različnih jedrnih funkcij	18
2.6	DAGSVM-graf	20
2.7	Primer klasifikacije s pomočjo teniškega turnirja	21
3.1	Enorazredni SVM	25
4.1	Ilustracija Metode 1	31
4.2	Barvni prostor RGB	32
4.3	Barvna BW-metoda	33
6.1	Hierarhična delitev učne množice	43
6.2	Merjenje podobnosti mask	44
6.3	Mera podobnosti za slike, zajete iz sosednjih rotacijskih kotov	45
6.4	Poravnava učnih množic	46
6.5	Zamaknjeno učno množico ponovno oštevilčimo	47
6.6	Izračun mere podobnosti za dvorazredni SVM	47
6.7	Hierarhija mask za primer slik duhca	48
6.8	Učne slike z masko	49
6.9	Hierarhija hiperkrogel	49
6.10	Primer klasifikacije za hierarhijo enorazrednega SVM-ja	50

6.11	Kombinacija eno- in dvorazrednih SVM-jev	50
6.12	Presek dveh mask	51
6.13	Primer, ko preseka dveh hiperkrogel ne računamo	52
6.14	Potrditveni postopek za dvorazredni SVM	53
7.1	Nekaj slik iz baze figuric	56
7.2	Primer učne množice za barvno BW-metodo	57
7.3	Nekaj slik razreda ozadja	57
7.4	Primeri testnih slik na različnih ozadjih	58
7.5	8 smeri premikanja slike	58
7.6	Energija lastnih vektorjev	65
7.7	Prvih 8 lastnih vektorjev glede na velikost lastnih vrednosti	66
7.8	Področja izbire slikovnih elementov	67
7.9	Hipoteze za duhca	68
7.10	Hipoteze za pelikana	68
7.11	Projekcije slik objektov v 2-D podprostor, določen z lastnimi vektorji	72
7.12	Projekcije slik objektov v 3-D podprostor, določen z lastnimi vektorji	72
7.13	Testne slike	77
7.14	Rezultati za pelikana pri različnih pragih glasovalne funkcije	78
7.15	Rezultati za črno ozadje	79
7.16	Rezultati za rdečkasto ozadje	80
7.17	Rezultati za realno ozadje	81
7.18	Objekti, ki sestavljajo bazo COIL-100	82
7.19	21 izbranih objektov iz COIL-100 baze	83
7.20	Primeri slik objekta obj 3 po normalizaciji velikosti in intenzitete	83
7.21	Testni slike za objekte iz COIL-100 baze	84
7.22	Objekti na testni sliki 1	86
7.23	Grafični prikaz rezultatov za prvo testno sliko (barvna BW-metoda)	86
7.24	Objekti na testni sliki 2	87
7.25	Grafični prikaz rezultatov za drugo testno sliko (barvna BW-metoda)	87
7.26	Grafični prikaz rezultatov za prvo testno sliko (hierarhični SVM)	88
7.27	Grafični prikaz rezultatov za drugo testno sliko (hierarhični SVM)	89

7.28	Število lastnih vektorjev in pripadajoče energije v primeru COIL-100 baze .	90
7.29	Projekcije slik objektov v 2-D podprostor, določen z lastnimi vektorji . . .	91
7.30	Projekcije slik objektov v 3-D podprostor, določen z lastnimi vektorji . . .	91
7.31	Uporabljene maske v primeru COIL-100 baze	92
7.32	Grafični prikaz rezultatov za testno sliko 1 (robustna PCA-metoda)	93
7.33	Grafični prikaz rezultatov za testno sliko 2 (robustna PCA-metoda)	94
7.34	Rezultati robustnega PCA na testni sliki 1 in pri vrednosti praga 0,5 . . .	95
7.35	Rezultati robustnega PCA na testni sliki 2 in pri vrednosti praga 0,5 . . .	95
8.1	Poravnava učnih slik	98
8.2	Maske, ki določajo lokalne značilke predmeta	100

Tabele

7.1	Rezultati za barvno BW-metodo pri raznolikem ozadju	59
7.2	Občutljivost barvne BW-metode na premik	60
7.3	Delež razpoznavne barvne BW-metode v odstotkih glede na premik	60
7.4	Rezultati za BW-metodo — skupno za vsa ozadja in vse premike	60
7.5	Rezultati za hierarhično metodo pri raznolikem ozadju	62
7.6	Občutljivost hierarhične SVM-metode na premik	63
7.7	Delež razpoznavne hierarhičnega SVM-ja v odstotkih glede na premik	63
7.8	Rezultati za hierarhično metodo — vse skupaj	63
7.9	Primerjava NN- in SVM-klasifikatorja za robustno PCA-metodo	69
7.10	Rezultati za robustno PCA-metodo pri raznolikem ozadju	71
7.11	Občutljivost robustne PCA-metode na premik	73
7.12	Delež razpoznavne za robustno PCA-metodo v odstotkih glede na premik	73
7.13	Povzetek rezultatov za robustno PCA-metodo, za vsak klasifikator posebej	73
7.14	Rezultati za vse tri metode v celoti	74
7.15	Primerjava deležev razpoznavne za vse tri metode v odstotkih glede na premik	74

Algoritmi

1	Faza razpoznavanja pri hierarhični SVM-metodi	54
---	---	----

Poglavje 1

Uvod

1.1 Pomen vizualne informacije

Vidno zaznavanje okolice ima v vsakdanjem življenju izredno velik pomen. Ljudje skoraj na vsakem koraku zgolj na podlagi vizualne informacije brez kakršnegakoli večjega napora razpoznavamo ne samo različnih predmetov, oseb, krajev in oznak, ampak tudi različna gibanja in celo občutke, ki jih vsiljujejo posamezni prizori. Vse to počnemo popolnoma spontano in se kompleksnosti celotnega procesa sploh ne zavedamo. Hkrati z opazovanjem in razpoznavanjem nadgrajujemo svoje vizualno znanje. Procesu učenja in razpoznavanja pri ljudeh potekata skoraj neprestano, to je vedno, kadar gledamo. Poleg tega smo ljudje sposobni kvalitetno opravljati naloge, povezane z gledanjem, v zelo ekstremnih pogojih, saj nismo preveč občutljivi na različne pogoje, ki nastajajo zaradi sprememb v okolici, kot so osvetlitev, sprememba zornega kota, delno prekrivanje predmetov in podobno.

Opisane lastnosti, ki jih pri vizualni razpoznavi zmore človek, bi želeli prenesti tudi na računalniško osnovan sistem. Veliko je namreč nalog, ki jih v vsakdanjem življenju opravljamo na podlagi vizualnih informacij. Težnja po čim večji avtomatizaciji procesov pronica že na vsa področja civilizacije. Vse več dela najraje prepuščamo strojem. Področje računalniškega vida se uveljavlja tam, kjer je za opravljanje dela potrebna sposobnost interpretacije vizualne informacije. Ker je z razvojem tehnologije in relativno nizkimi cenami različnih kamer in fotoaparátov vse več stvari nadzorovanih in dokumentiranih prav v obliki slik in video posnetkov, se potreba po sistemih, ki gledajo, vsak dan povečuje.

Iskanje določenega predmeta ali osebe v video sekvenci, analiza posnetkov športnih iger, avtomatsko nakupovanje s pomočjo robota v samopostrežni trgovini, opazovanje prometne signalizacije pri avtonomni vožnji in nadzorovanje gibanja so le nekatere naloge v množici aplikacij, kjer je analiza vizualne informacije bistvenega pomena.

1.2 Problemi pri razpoznavanju objektov na podlagi vizualne informacije

V nasprotju s človekom, ki z lahkoto opravlja interpretacijo vizualne informacije, pa je to precej težka in kompleksna naloga za stroj. Težave, na katere naletimo pri razpoznavanju, so vezane predvsem na spremembe v okolici, ki vplivajo na izged prizora. Te so:

- zakrivanja med objekti,
- sprememba ozadja,
- različna osvetlitev,
- različen zorni kot opazovanja,
- različna skala, ki je posledica različne oddaljenosti predmeta od opazovalca,
- različne vrste šumov, ki nastajajo v procesih zajemanja, zgoščevanja, prenašanja ali branja slik.

V konkretnem delu smo se osredotočili na problem razpoznavanja togih objektov na različnih ozadjih, opazovanih pri različnih zornih kotih.

1.3 Metode računalniškega vida za razpoznavanje 3-D objektov

Na področju računalniškega vida sta se uveljavila dva pristopa pri reševanju problema razpoznavne 3-D objektov. Prvi pristop zahteva gradnjo 3-D modela objekta. Primer

takega pristopa je modeliranje s pomočjo superkvadrikov, ki uporablja globinske slike. Drugi pristop, glediščno osnovani (ang. *appearance based*), temelji na vizualnem izgledu objektov na slikah.

Bistvo glediščno osnovanega pristopa je, da vidno informacijo, sliko, obravnavamo kot celoto. Pri tem ne ločujemo posameznih značilnosti, kot so oblika, tekstura, osvetlitev, odbojne lastnosti površine objekta in njegov položaj v prostoru. Celoten proces modeliranja in razpoznavanja poteka v domeni 2-D slik.

Razpoznavanje spominja na ujemanje (ang. *matching*), saj navadno poteka tako, da testno sliko uvrstimo v skupino, ki je testni sliki glede na izgled najbližja. Medtoda podpornih vektorjev, metoda glavnih komponent, kanonična korelacijska analiza, metoda neodvisnih komponent in linearna diskriminantna analiza so nekatere izmed pogosto uporabljenih tehnik glediščno osnovanega pristopa.

Učenje in razpoznavanje sta v osnovnih izvedbah metod, osnovanih na izgledu, navadno ločena. Vhod v učno fazo je nabor slik, ki naj bi čim bolje predstavljal predmet, ki se ga učimo. Učne podatke pripravimo glede na potrebe razpoznavanja. Če bomo predmet razpoznavali pod različnimi osvetlitvami, je smiselno, da učna množica vsebuje slike, ki prikazujejo objekt pod različnimi osvetlitvami, če ga bomo poskušali razpoznavati iz različnih zornih kotov, potem tudi učne primere kreiramo tako, da prikazujejo predmet iz različnih pogledov, itd. Izhod učnega procesa je lahko drugačna, bolj kompaktna predstavitev učnih primerov (metoda glavnih komponent) ali pa kar funkcija, ki jo uporabljamo za klasifikacijo (metoda podpornih vektorjev). Ko je kriterijska funkcija določena, nove podatke enostavno razvrščamo glede na njeno vrednost.

Obstajajo tudi pristopi, kjer se faza učenja in faza razpoznavanja izmenjujeta. To so t.i. inkrementalne metode, ki omogočajo postopno gradnjo modelov [23] in se zdijo bližje človekovemu načinu učenja.

V delu smo se osredotočili na metodo podpornih vektorjev in metodo glavnih komponent.

1.4 Cilji naloge

Pri reševanju nalog računalniškega vida moramo dopustiti tudi možnost, da slika ali njen del ne vsebuje nobenega izmed znanih objektov. Pomembno je torej oboje: razpoznati objekt na sliki, ko se ta na sliki nahaja, in “ne razpoznati” objekta, ko ga na sliki ni. Pri vrednotenju procesa razpoznavne merimo oboje: stopnjo oz. delež pravilno razpoznanih objektov na sliki — *pravilno pozitivni*, (ang. *true positives*, TP), kot tudi delež razpoznanih objektov na sliki, ki to niso — *napačno pozitivni*, (ang. *false positives*, FP). Izkaže se, da je težko zmanjšati stopnjo FP in pri tem ohraniti TP ter obratno, povečati TP, ne da bi pri tem povečali FP. Slike, ki ne prikazujejo znanih objektov, lahko predstavljajo poseben razred — razred ozadja. Zelo težko je določiti značilne primere za razred ozadja, saj bi ta moral teoretično vsebovati vse možne slike ozadja. Razred ozadja se zato pogosto določi v več iteracijah. V vsaki iteraciji se učni množici ozadja dodajo slike ozadja, ki so bile v procesu razpoznavne napačno razpoznane kot objekt.

V nalogi poskušamo razviti glediščno osnovano metodo, za lociranje in razpoznavanje togih objektov na poljubnem ozadju, ki bi čim bolj uspešno zmanjšala visoko stopnjo napačno pozitivnih razpoznav, ne da bi pri tem zmanjšala stopnje pravilno pozitivnih razpoznav.

1.5 Pregled sorodnih del

S problemom razpoznavanja objektov na poljubnem ozadju na podlagi metode podpornih vektorjev se je ukvarjal Roobaert [18]. Predlagal je postopek pedagoškega učenja, t.i. BW-metodo, ki poskuša odločitveno funkcijo izračunati tako, da zmanjša vpliv slikovnih elementov, ki pripadajo ozadju. Predlagani postopek dobro deluje v primeru sivinskih slik. Problem, ki se pojavi pri takem načinu razpoznavanja je velika stopnja napačno pozitivnih.

Tudi pristop A. Selinger [21, 22] temelji na glediščno osnovanih metodah. Za predstavitev 3-D objektov predlaga štirinivojski sistem. Na prvem nivoju na osnovi izračunanih robov objekte predstavi s krivuljnimi odseki. Na drugem nivoju poveže krivuljne odseke iz prvega nivoja v kontekstne regije (ang. *context patches*). Tretji nivo pred-

stavljajo različni 2-D pogledi na posamezen predmet, ki na četrtem nivoju skupaj tvorijo predstavitev 3-D objekta. Zanimivo je, da se četrtega nivoja pri razpoznavanju poslužimo le pri nejasnih primerih. V negotovih situacijah robot obrne predmet, ga “pogleda” še iz druge strani, tvori novo hipotezo in jo primerja s prvo. V primeru skladnega rezultata objekt uspešno razpozna, v nasprotnem primeru generira novo hipotezo iz novega pogleda. Z načinom uporabe četrtega nivoja se približuje človeškemu načinu razpoznavanja. Tipično za človeka je namreč, da če nekega predmeta ne razpozna iz enega pogleda, poskuša z razpoznavo iz drugih pogledov. Pristop A. Selinger se od drugih razlikuje tudi v številu prostostnih stopenj, s katerimi predstavi posamezen objekt. Medtem ko večina sistemov objekte poslika okrog ene rotacijske osi, njen pristop temelji na informaciji, ki jo dobimo, če objekt poslikamo iz vseh možnih smeri — na diskretnih točkah hemisfere.

Problemi, kjer na sliki poskušamo poiskati vse primere objektov, ki spadajo v neko splošno skupino, kot so obrazi, ljudje itd. so sicer zelo podobni tistim, ki med posameznimi objekti tudi ločujejo, vendar imajo specifične karakteristike. Problemi te vste so dvorazredni, ločujejo le med razredom učenega objekta in ostalim. Pri nalogi iskanja in razpoznavanja različnih togih 3-D objektov ne moremo uporabiti istih tehnik, lahko pa se poslužujemo nekaterih zanimivih idej, ki so bile uspešno uporabljene v dvorazrednih primerih. Viola in Jones sta se v delu [9] osredotočila na problem iskanja obrazov. Rešitev, ki jo predlagata, sloni na AdaBoost algoritmu. Zanimiva pa je zaporedna uporaba posameznih klasifikatorjev s specifičnimi lastnostmi. Najprej uporabimo enostavne klasifikatorje, ki morajo biti zgrajeni tako, da iz nadaljnje obdelave izločijo čim več slik, ki z veliko verjetnostjo ne vsebujejo iskanega objekta. Vrednost funkcije klasifikatorjev na prvih nivojih izračunamo zelo hitro. Ta lastnost omogoča, da malo časa porabimo na delih slik, ki predstavljajo ozadje. Klasifikatorji na zadnjih nivojih so kompleksnejši in posledično tudi časovno bolj potratni. Vendar to ni bistvenega pomena, saj se izvajajo le na tistih primerih, kjer lahko s precej veliko verjetnostjo trdimo, da določen del slike vsebuje znan objekt. Tudi naš hierarhičen sistem razpoznavanja, ki je podrobneje predstavljen v 6. poglavju spominja na tak sistem: Iskanje posameznega objekta na zadnjih nivojih hierarhije izvajamo le, če potrditev uspe na predhodnih nivojih.

Tudi C. Papageorgiou in soavtorji v [16] obravavajo problem z dvema razredoma: pešci in nepešci. Namen sistema je opozarjati voznika na pešce. Pešči se pojavljajo na

poljubnem ozadju in se med seboj zelo razlikujejo. Predstavitev razreda, ki bi temeljila na vrednosti slikovnih elementov slike pešca, ne bi zadostila zahtevam za uspešno razpoznavanje. Pri predstavitvi posameznih razredov je namreč pomembno, da je variabilnost znotraj razreda majhna, medrazredna variabilnost pa čim večja. Avtorji dela izvedejo nad slikami Haarove transformacijo, ki jih računajo za različne barvne kanale, za različne smeri (horizontalno, vertikalno in diagonalno) ter pri različnih velikostih slik. Za učenje razpoznavanja nad tako pridobljenimi vektorji značilk uporabijo metodo podpornih vektorjev.

Matas s soavtorji v delih [12, 15] problem ozadja ter morebitnih zakrivanj rešuje s pomočjo lokalnih značilnosti. Vsak predmet predstavi z vrsto razločevalnih regij (ang. *distinguish regions*). Razločevalne regije imajo vrsto atraktivnih lastnosti kot so: invariantnost na affine transformacije, na perspektivo in na monotone transformacije intenzitete slik, regije lahko hitro izračunamo, pri uporabi razločevalnih regij ni potrebno glajenje slik kar omogoča, tako fino kot grobo predstavitev strukture predmetov. Tako razpoznavanje lahko izvajamo v realnem času. Lepa stran te metode je tudi, da tolerira relativno veliko stopnjo zakrivanj, različna ozadja in šum. Klasifikacija poteka tako, da na objektu, ki ga razpoznavamo poiščemo razločevalne regije in jih primerjamo z regijami znanih objektov. Težave se pojavijo, če objekt nima področji, ki bi jih lahko uporabili kot razločevalne regije. Primer takega objekta je enobarvna žoga. Takih objektov na ta način ne moremo razpoznavati.

Ker si različna ozadja lahko razlagamo tudi kot neke vrste šum na sliki, lahko kot relevantno metodo omenimo tudi način robustnega računanja PCA-koeficientov, ki ga predlagata Leonardis in Bischof [10]. Metoda osnovnih komponent omogoča v procesu razpoznave tudi rekonstrukcijo objekta. Robustni PCA uporablja večje število hipotez izračunanih na različnih podmnožicah na sliki. S pomočjo izračunanih rekonstrukcij najprej izloči slikovne elemente ozadja, nato pa izvede proces razpoznave. Tega načina smo se poslužili tudi v nalogi in ga podrobneje opisuje 5. poglavje.

1.6 Struktura naloge

V drugem in tretjem poglavju naloge opisujemo teoretične osnove dvorazrednih in enorazrednih SVM-jev. Matematični predstavitvi problema sledijo tri poglavja, od katerih vsako opisuje drugačen pristop k reševanju problema poljubnega ozadja, in sicer: poglavje 4 opisuje BW-metodo, ki temelji na pedagoškem učenju. Opisana je tudi naša nadgradnja BW-metode za primer barvnih slik. V poglavju 5 smo predstavili PCA-metodo ter njeno robustno različico računanja PCA-koeficientov. V poglavju 6 je opisana metoda, ki smo jo predlagali za razpoznavanje objektov na poljubnem ozadju. Temelji na hierarhiji eno- in dvorazrednih SVM-jev. Slednja dva opisana pristopa uporabljata pri razpoznavi podmnožice slikovnih elementov na slikah.

Vse tri opisane metode smo testirali na slikah z različnimi ozadji. Izvedli smo tudi teste, kjer je bilo potrebno objekt ne le klasificirati, ampak poiskati tudi njegovo lokacijo na sliki. Eksperimenti in rezultati so podani v 7. poglavju. Delo zaključimo z diskusijo ter smernicami za nadaljnje delo.

Poglavje 2

Metoda podpornih vektorjev

Metoda podpornih vektorjev (ang. *Support Vector Machine*, SVM) je v osnovi binarni, linearni klasifikator. Med dvema razredoma ločuje s pomočjo hiperravnine, določene s funkcijo $f(x)$, ki deli hiperprostor na dva dela.

2.1 Formulacija problema

Podanih imamo N točk \mathbf{x}_i , $i = 1, 2, \dots, N$, $\mathbf{x}_i \in \mathbb{R}^M$. Vsaka točka pripada natanko enemu izmed dveh razredov $y_i \in \{-1, 1\}$. Pari $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ tvorijo učno množico. Predpostavljamo, da je učna množica linearno ločljiva, kar pomeni, da obstaja hiperravnina, ki razdeli prostor na dva dela tako, da so točke, ki pripadajo istemu razredu, na isti strani hiperravnine. Za hiperravnino $\langle \mathbf{x} \cdot \mathbf{w} \rangle + b = 0$, določeno z vektorjem $\mathbf{w} \in \mathbb{R}^M$ in konstanto $b \in \mathbb{R}$, torej velja:

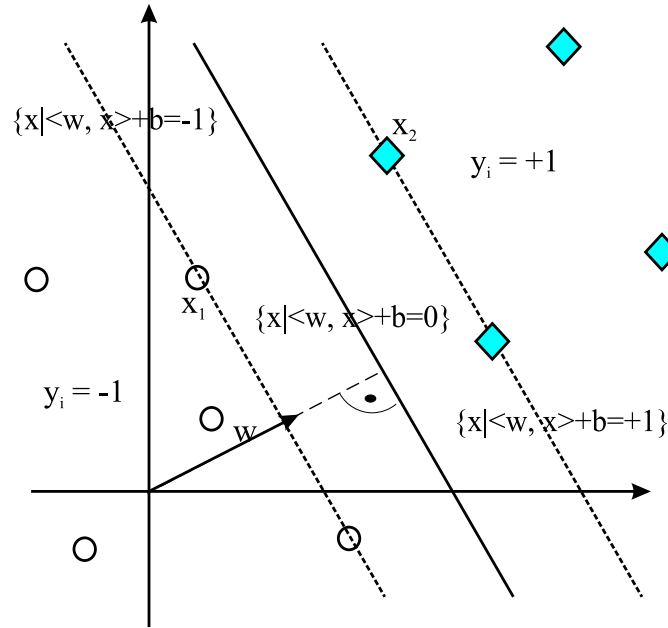
$$\forall i, \quad y_i \cdot (\langle \mathbf{x}_i, \mathbf{w} \rangle + b) \geq 0.$$

Odločitvena funkcija $f(\mathbf{x})$ ima obliko:

$$f(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b). \quad (2.1)$$

Če skaliramo \mathbf{w} in b tako, da za točke, ki so hiperravnini najbližje, velja $|\langle \mathbf{w}, \mathbf{x}_i \rangle + b| \geq 1$, dobimo kanonično obliko (\mathbf{w}, b) hiperravnine.

Optimalna razločevalna hiperravnina (ang. *Optimal Separating Hyperplane*, OSH) je tista hiperravnina, ki maksimizira razdaljo med točkami, ki so hiperravnini najbližje. OSH ima med vsemi razločevalnimi hiperravninami največjo sposobnost generalizacije [5]. Metoda podpornih vektorjev temelji na OSH-ju (slika 2.1, enačba 2.2).



Slika 2.1: Grafični prikaz OSH-ja za enostaven dvodimenzionalni primer. Predstavnike prvega razreda označujejo krogi, predstavnike drugega pa modri rombi. Polna črta med črtkanima črtama predstavlja OSH. Črtkani črti sta meji pasu med dvema razredoma. \mathbf{x}_1 in \mathbf{x}_2 sta podporna vektorja (ang. *support vectors*).

$$\begin{aligned}
 \langle \mathbf{w}, \mathbf{x}_1 \rangle + b &= +1 \\
 \langle \mathbf{w}, \mathbf{x}_2 \rangle + b &= -1 \\
 \Rightarrow \langle \mathbf{w}, (\mathbf{x}_1 - \mathbf{x}_2) \rangle &= 2 \\
 \Rightarrow \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, (\mathbf{x}_1 - \mathbf{x}_2) \right\rangle &= \frac{2}{\|\mathbf{w}\|}
 \end{aligned} \tag{2.2}$$

Če želimo maksimizirati širino pasu, ki ločuje oba razreda, moramo minimizirati $\|\mathbf{w}\|$. Problem iskanja OSH-ja zapišemo kot kvadratni optimizacijski problem:

$$\underset{\mathbf{w} \in \mathbb{R}^M, b \in \mathbb{R}}{\text{minimiziraj}} \quad \tau(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.3)$$

$$\text{glede na omejitve} \quad y_i \cdot (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 \quad i = 1, \dots, N \quad (2.4)$$

Opazimo lahko, da omejitev (2.4) zagotavlja, da funkcija $f(\mathbf{x}_i)$ zavzema vrednost $+1$ pri vrednosti $y_i = +1$ in -1 pri $y_i = -1$. Lahko bi sicer trdili, da je dovolj, če v enačbi (2.4) namesto ≥ 1 zapišemo > 0 , vendar bi bilo v takem primeru nesmiselno minimizirati dolžino $\|\mathbf{w}\|$. Neenačba z omejitvijo > 0 je še vedno izpolnjena, če neko rešitev (\mathbf{w}, b) skaliramo z λ , $0 < \lambda < 1$. Če pa namesto ničle na desni strani neenačbe (2.4) uporabimo katerokoli pozitivno število, npr. 1, fiksiramo skaliranje \mathbf{w} -ja. Dejansko imamo v primeru > 0 neskončno število rešitev, ki se medseboj razlikujejo le za nek faktor \mathbf{w} -ja [20, 5].

2.2 Določanje vezanega ekstrema

Osnovni optimizacijski problem (2.3) z omejitvami (2.4) zapišemo kot Lagrangeovo funkcijo in prevedemo na ustrezni dualni problem.

Naj $\alpha_i \geq 0, i = 1, \dots, N$ označujejo Lagrangeove multiplikatorje. Lagrangeovo funkcijo $L(\mathbf{w}, b, \boldsymbol{\alpha})$:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i (y_i (\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1) \quad (2.5)$$

moramo minimizirati glede na spremenljivke \mathbf{w} , b in α_i .

Karush-Kuhn-Tuckerjev (KKT) izrek zahteva, da rešitev zadošča naslednjima pogojema:

$$\frac{\delta}{\delta b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{0} \quad (2.6)$$

$$\frac{\delta}{\delta \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{0} \quad (2.7)$$

Iz enačb 2.6 in 2.7 sledi:

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (2.8)$$

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \quad (2.9)$$

Rešitev za \mathbf{w} (2.9) zapišemo s podmnožico učne množice tistih učnih primerov, za katere velja $\alpha_i \neq 0$. Te učne primere imenujemo *podporni vektorji* (ang. *Support Vector*, SV). Od tu izvira tudi poimenovanje metode: metoda podpornih vektorjev.

Iz KKT pogojev

$$\alpha_i (y_i (\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1) = 0 \quad i = 1, \dots, N \quad (2.10)$$

sledi, da podporni vektorji ležijo ravno na meji. Vsi ostali učni primeri so za rešitev nepomembni.

Če izraza (2.8) in (2.9) vstavimo v Lagrangeovo enačbo (2.5), dobimo dualni optimizacijski problem $L_D(\alpha)$ z omejitvami. Za izračun SVM-jev uporabljamo L_D :

$$\underset{\alpha \in \mathbb{R}^n}{\text{maksimiziraj}} \quad L_D(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \quad (2.11)$$

$$\text{glede na omejitve } \alpha_i \geq 0 \quad i = 1, \dots, N \quad (2.12)$$

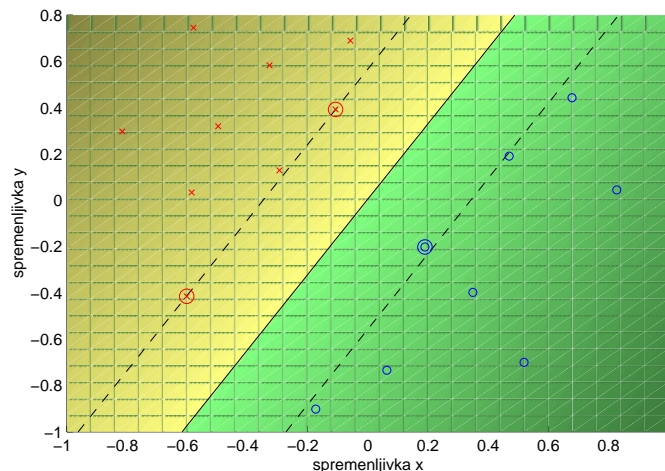
$$\text{in } \sum_{i=1}^N \alpha_i y_i = 0 \quad (2.13)$$

\mathbf{w} izračunamo po enačbi (2.9). Vrednost b določimo s pomočjo podpornih vektorjev pri pogoju $y_i \cdot (\langle \mathbf{x}_i, \mathbf{w} \rangle + b) = 1$. Izračunani b -ji se za različne podporne vektorje lahko med seboj nekoliko razlikujejo. b , ki ga uporabljamo pri klasifikaciji, navadno izračunamo kot povprečje b -jev, izračunanih pri različnih podpornih vektorjih.

Odločitveno funkcijo (2.1) z uporabo enačbe (2.9) sedaj zapišemo:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^N y_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b\right). \quad (2.14)$$

Primer klasifikacije z uporabo Metode podpornih vektorjev je prikazan na sliki 2.2.



Slika 2.2: Primer SVM-klasifikatorja. Na sliki je področje prvega razreda obarvano z rumeno, področje drugega razreda pa z zeleno barvo. Bolj je barva temna, večja je vrednost izraza $|\langle \mathbf{w}, \mathbf{x}_i \rangle + b|$. Vidimo, da vrednost narašča z oddaljevanjem od OSH-ja. Učni podatki so označeni z rdečimi križci oz. modrimi krogci. Podporni vektorji so v primeru modrih oznak dvojni krogci, v primeru rdečih pa z obkroženim križcem.

2.3 Mehka meja (ang. *soft margin*)

V prejšnjem poglavju smo matematično ozadje SVM-jev prikazali na linearno ločljivi učni množici. Pri reševanju problemov naletimo tudi na množice, ki niso linearno ločljive. Že določena stopnja šuma lahko povzroči, da se sicer linearno ločljive množice med seboj prekrivajo. V takih primerih lahko pogoj (2.4) nekoliko omilimo. V ta namen uvedemo dodatne spremenljivke ξ_i , $i = 1, \dots, N$ (ang. *slack variable*), ki dopuščajo tudi napačno klasifikacijo v fazi učenja. Omejitve iz enačbe (2.4) zapišemo kot:

$$\begin{aligned} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) &\geq 1 - \xi_i \quad i = 1, \dots, N \\ \xi_i &\geq 0 \quad i = 1, \dots, N \end{aligned} \tag{2.15}$$

Če želimo, da bo klasifikator dobro generaliziral, moramo pravilno uravnotežiti širino roba $\|w\|$ in vsoto kazenskih spremenljivk $\sum_i \xi_i$. Nalogo uravnoteženja opravlja parameter C . Optimizacijski problem lahko zapišemo kot:

$$\underset{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}}{\text{minimiziraj}} \quad \tau(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \left(\sum_{i=1}^N \xi_i \right)^k \quad (2.16)$$

$$\begin{aligned} \text{glede na omejitve} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, \dots, N \end{aligned} \quad (2.17)$$

Parameter C ($C > 0$) definira uporabnik. Večji kot je C , bolj kaznujemo napačno klasificirane primere — manjše klasifikacijske napake dopuščamo. Za katerokoli pozitivno število k je problem zapisan v obliki (2.16) še vedno konveksen. Če pa velja $k = 1$ ali $k = 2$, imamo kvadratično optimizacijsko nalogo. Pri vrednosti $k = 1$, se pojavi še dodatna lepa lastnost: v dualni obliki izginejo tako kazenske spremenljivke ξ_i kot tudi Lagrangeovi multiplikatorji α_i .

Lagrangeova funkcija ima v primeru mehkih mej naslednjo obliko:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_i^N \xi_i - \sum_i^N \alpha_i (y_i(\langle x_i, \mathbf{w} \rangle) - 1 + \xi_i) - \sum_i^N \mu_i \xi_i \quad (2.18)$$

μ_i so Lagrangeovi multiplikatorji, ki uveljavljajo pozitivnost ξ_i . KKT-pogoji za primarno Lagrangeovo funkcijo so:

$$\frac{\delta}{\delta \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \mathbf{w} - \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i = 0 \quad (2.19)$$

$$\frac{\delta}{\delta b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = - \sum_{i=1}^N \alpha_i y_i = 0 \quad (2.20)$$

$$\frac{\delta}{\delta \xi_i} L(\mathbf{w}, b, \boldsymbol{\alpha}) = C - \alpha_i - \mu_i \quad (2.21)$$

$$y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1 + \xi_i \geq 0 \quad (2.22)$$

$$\xi_i \geq 0 \quad (2.23)$$

$$\alpha_i \geq 0 \quad (2.24)$$

$$\mu_i \geq 0 \quad (2.25)$$

$$\alpha_i(y_i(\langle \mathbf{x}_i, \mathbf{w} \rangle + b) - 1 + \xi_i) = 0 \quad (2.26)$$

$$\mu_i \xi_i = 0 \quad (2.27)$$

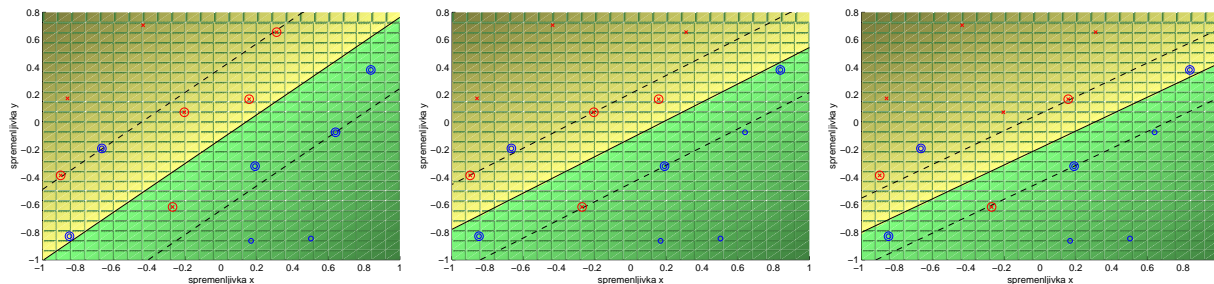
Dualno obliko problema OSH z mehкими mejami zapišemo kot:

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\text{maksimiziraj}} \quad L_D(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \quad (2.28)$$

glede na omejitve $0 \leq \alpha_i \leq C \quad i = 1, \dots, N$

$$\text{in} \quad \sum_{i=1}^N \alpha_i y_i = 0 \quad (2.29)$$

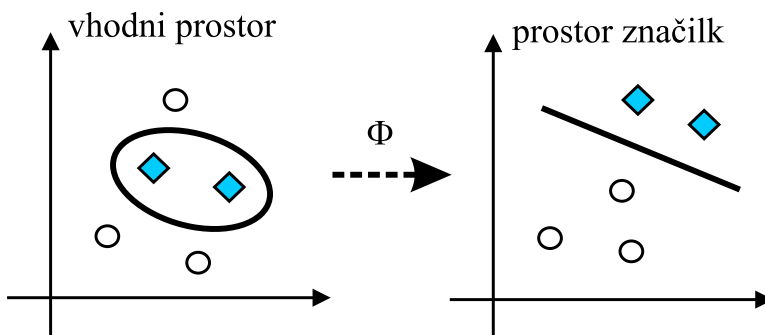
Edina razlika med OSH-jem s strogo mejo in OSH-jem z mehko mejo je dodatna omejitev C za Lagrangeove spremenljivke. Odločitvena funkcija ima enako obliko kot funkcija v primeru stroge meje (2.14). b izračunamo iz (2.26) in (2.27). Iz (2.19) in (2.27) sledi, da je $\xi_i = 0$, če je $\alpha_i < C$. b dobimo tako, da vzamemo neko α_i , $0 < \alpha_i < C$ in jo vstavimo v enačbo (2.26), pri pogoju $\xi_i = 0$. Primer OSH-ja z mehko mejo na linearno neločljivi množici je prikazan na sliki 2.3.



Slika 2.3: Na sliki vidimo primer OSH-ja za linearno neločljivo množico pri $C=2$, $C=50$ in $C=700$. Večji kot je C , bolj kaznujemo napačno klasificirane vzorce. Večji kot je C , ožji je rob.

2.4 Uporaba jedra

Klasifikacijske funkcije iz prejšnjega poglavja so linearne. V splošnem seveda obstaja vrsta problemov, za katere linearni klasifikator ni primeren; npr. če podatki niso linearno ločljivi. V takih primerih lahko uporabimo jedrne funkcije (ang. *kernel function*). Z uporabo jedrnih funkcij preslikamo podatke iz vhodnega prostora (ang. *input space*) v nek višjedimenzijski prostor značilk (ang. *feature space*). V prostoru značilk nato konstruiramo OSH, ki je v vhodnem prostoru dejansko nelinearna odločitvena funkcija. Ideja preslikave iz vhodnega prostora v prostor značilk je prikazana na sliki 2.4.



Slika 2.4: Ideja jedra: podatke iz vhodnega prostora preslikamo v prostor značilk, kjer poiščemo OSH. Posledica preslikave je nelinearna meja med razredoma v vhodnem prostoru.

Če npr. želimo izračunati odločitveno funkcijo, ki ima v vhodnem prostoru obliko polinoma druge stopnje, moramo najprej ustrezno konstruirati prostor značilk. Preslikavo $\Phi(\mathbf{x})$, ki preslika M dimenzijski vhodni prostor v $M' = \frac{M(M+3)}{2}$ dimenzijski prostor značilk, definiramo kot:

$$\begin{array}{lll} z_1 = x_1 & z_{M+1} = (x_1)^2 & z_{2M+1} = x_1 \cdot x_2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ z_M = x_M & z_{2M} = (x_M)^2 & z_{M'} = x_M \cdot x_{M-1} \end{array}$$

kjer je $\mathbf{x} = (x_1, \dots, x_M)$. Funkcija, ki določa OSH v prostoru značilk, je v vhodnem prostoru polinomska funkcija reda 2.

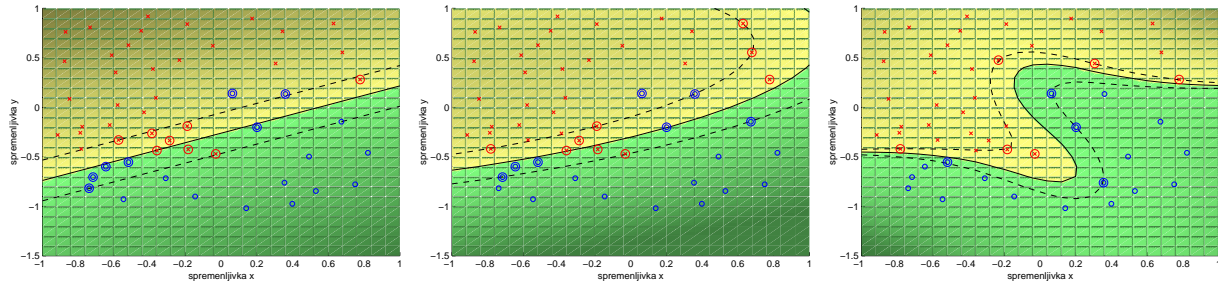
Število dimenzij v takih primerih zelo naraste. Npr. pri 200-dimezijskem problemu v vhodnem prostoru dobimo ob uporabi polinomske preslikave 4. oz. 5. stopnje prostor značilk, katerega dimenzijo izražamo v bilijonih. Pri računanju s tako razsežnim prostorom bi kaj hitro naleteli na težave, saj je tako velik problem tehnično neobvladljiv. Tako imenovano “prekletstvo dimenzij” pa pri SVM-jih elegantno zaobidemo. Če si pozorno ogledamo enačbi (2.12) in (2.14), vidimo, da neposredna preslikava v prostor značilk sploh ni potrebna. Zadošča že, če v prostoru značilk izračunamo skalarni produkt med podpornim vektorjem in vektorjem iz prostora značilk. V formulah vektorji iz prostora značilk vedno nastopajo v obliki skalarnega produkta, torej jih ni nikoli potrebno izraziti eksplicitno.

Skalarni produkt $\langle \mathbf{a}, \mathbf{b} \rangle$ zapišemo kot $k(\mathbf{a}, \mathbf{b})$, kar ustreza skalarnemu produktu v prostoru značilk.

$$k(\mathbf{a}, \mathbf{b}) = \langle \Phi(\mathbf{a}), \Phi(\mathbf{b}) \rangle \quad (2.30)$$

$$\Phi : \mathbb{R}^N \rightarrow \mathcal{H} \quad (2.31)$$

To zamenjavo poznamo pod imenom trik s pomočjo jedra (ang. *kernel trick*). Bistvo trika je, da nam v prostor značilk ni potrebno direktno preslikati vsake točke posebej. Dovolj je, da poznamo skalarni produkt v prostoru značilk, torej funkcijo k .



Slika 2.5: Primer uporabe različnih jedrnih funkcij, ki zadoščajo Mercerjevemu pogoju. $C=100$ za vse primere. Uporabljene jedrne funkcije od leve proti desni: linearna, polinomska stopnje 2 in polinomska jedrna funkcija stopnje 3.

Z uporabo jedrnih funkcij k , ki zadoščajo Mercerjevemu pogoju [3], lahko konstruiramo nelinearno odločitveno funkcijo oblike:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b, \quad (2.32)$$

ki je enakovredna linearni odločitveni funkciji v ustreznem prostoru značilk. Koeficiente α_i iz izraza (2.32) izračunamo tako, da rešimo kvadratni optimizacijski problem (2.33).

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\text{maksimiziraj}} \quad L_D(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (2.33)$$

$$\text{glede na omejitve } 0 \leq \alpha_i \leq C \quad i = 1, \dots, N$$

$$\text{in } \sum_{i=1}^N \alpha_i y_i = 0$$

Odločitvena funkcija pa ima obliko:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^N y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b\right). \quad (2.34)$$

Nekaj najbolj znanih jedrnih funkcij [3]:

polinom stopnje p : $k(\mathbf{a}, \mathbf{b}) = (\mathbf{a} \cdot \mathbf{b} + 1)^p$,

Gaussova radialna bazna funkcija (RBF): $k(\mathbf{a}, \mathbf{b}) = e^{-\|\mathbf{a}-\mathbf{b}\|^2/2\sigma^2}$,

dvonivojska sigmoidna nevronska mreža: $k(\mathbf{a}, \mathbf{b}) = \tanh(\kappa \mathbf{a} \cdot \mathbf{b} - \delta)$.

2.5 Metoda podpornih vektorjev za klasifikacijo v več razredov

Metoda SVM je v osnovi binarni klasifikator. Za velik del klasifikacijskih problemov delitev v samo dva razreda ne zadošča. Obstaja kar nekaj načinov, kako uporabiti tehniko SVM pri klasificiranju v primeru več razredov. V nadaljevanju je naštetih in opisanih nekaj najbolj pogostih [8]. Glede na rezultate testov [8] dosežemo najboljše rezultate z metodo DAGSVM.

2.5.1 Eden proti vsem ostalim

Metoda *eden proti vsem ostalim* (ang. *one against all*) je ena prvih metod za klasifikacijo v več razredov s pomočjo SVM-jev. Metoda zahteva, da zgradimo k SVM modelov, k je število razredov. i -ti SVM, $i = 1, \dots, k$, ločuje i -ti razred od vseh ostalih. Takšen princip vodi do k odločitvenih funkcij:

$$\begin{aligned} &\langle \mathbf{w}^1, \phi(\mathbf{x}) \rangle + b^1 \\ &\quad \cdot \\ &\quad \cdot \\ &\quad \cdot \\ &\langle \mathbf{w}^k, \phi(\mathbf{x}) \rangle + b^k \end{aligned}$$

Točko \mathbf{x} uvrstimo v razred, ki da največjo vrednost odločitvene funkcije:

$$\text{razred } \mathbf{x} \equiv \underset{i=1, \dots, k}{\operatorname{argmax}} \langle \mathbf{w}^i, \phi(\mathbf{x}) \rangle + b^i.$$

2.5.2 Eden proti enemu

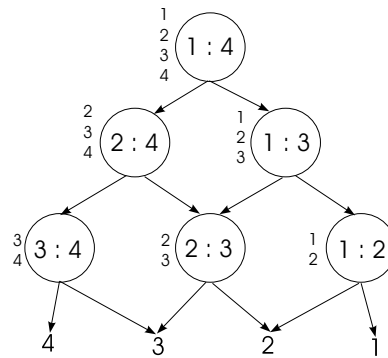
Metoda *eden proti enemu* (*ang. one against one*) je osnovana na $\frac{k(k-1)}{2}$ SVM-klasifikatorjih. Vsak klasifikator izračunamo iz podatkov dveh izmed k razredov. Pri tem nastopa i -ti razred v kombinaciji z vsemi drugimi $(k - 1)$ razredi. Za podatke i -tega in j -tega razreda dobimo odločitveno funkcijo:

$$\langle \mathbf{w}^{i,j}, \phi(\mathbf{x}) \rangle + b^{i,j}$$

Pri klasifikaciji vektorja \mathbf{x} različni klasifikatorji uvrščajo \mathbf{x} v različne razrede. Končno odločitev o pripadnosti \mathbf{x} -a enemu izmed razredov lahko določimo na osnovi števila zbranih glasov. Če odločitvena funkcija $\text{sign}(\langle \mathbf{w}^{i,j}, \phi(\mathbf{x}) \rangle + b^{i,j})$ pravi, da \mathbf{x} pripada i -temu razredu, prištejemo en glas i -temu razredu, sicer prištejemo en glas j -temu razredu. \mathbf{x} uvrstimo v razred, ki je dobil največ glasov.

2.5.3 DAGSVM

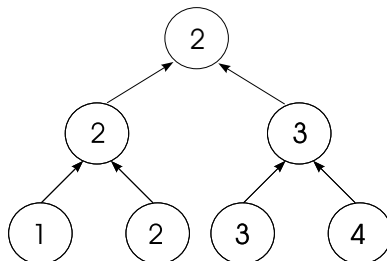
Tudi pri DAGSVM (*ang. Directed Acyclic Graph Support Vector Machine*) zgradimo $\frac{k(k-1)}{2}$ SVM-jev na enak način kot pri strategiji eden proti enemu. S SVM-ji zgradimo binarni usmerjen aciklični graf, ki ima $\frac{k(k-1)}{2}$ vozlišč in k listov. Testiranje začnemo v korenu grafa in končamo pri listih, ki določajo, kateremu razredu pripada \mathbf{x} [17]. Smer potovanja po grafu določa odločitvena funkcija vsakega posameznega vozlišča (slika 2.6).



Slika 2.6: DAGSVM graf

2.5.4 Teniški turnir

Tudi tu zgradimo binarni graf, ki ima na najnižjem nivoju k vozlišč. Vsako vozlišče predstavlja en razred. Po dve in dve vozlišči primerjamo med seboj na podlagi ustrezne odločitvene funkcije. $\frac{k(k-1)}{2}$ odločitvenih funkcij je tudi v tem primeru zgrajenih po principu eden proti enemu. Tisti razredi, za katere so odločitvene funkcije glasovale, se uvrstijo na višji nivo. Postopek ponavljamo, dokler v koren drevesa ne dobimo le en razred - to je razred, v katerega klasificiramo (slika 2.7).



Slika 2.7: Primer klasifikacije s pomočjo teniškega turnirja

2.5.5 k-razredni SVM

Problem uvrščanja v k -razredov pa lahko rešimo tudi direktno, s preoblikovanjem zapisa 2.16 tako, da nam omogoči izračun klasifikatorja za več razredni SVM. [24]

$$\underset{\mathbf{w} \in \mathcal{H}, b \in \mathbb{R}, \xi^r \in \mathbb{R}^M}{\text{minimiziraj}} \quad \tau(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2} \sum_{r=1}^k \|\mathbf{w}_r\|^2 + \mathbf{C} \sum_{i=1}^M \sum_{\mathbf{r}_i \neq \mathbf{y}_i} \xi_i^{\mathbf{r}_i} \quad (2.35)$$

$$\begin{aligned} \text{glede na omejitve} \quad & \langle \mathbf{w}_{\mathbf{y}_i}, \mathbf{x}_i \rangle + b_{y_i} \geq \langle \mathbf{w}_{\mathbf{r}}, \mathbf{x}_i \rangle + b_r + 2 - \xi_i^{\mathbf{r}} \\ & \xi_i^{\mathbf{r}} \geq 0 \quad i = 1, \dots, N \quad m \in \{1, \dots, k\} \setminus y_i \end{aligned}$$

Odločitvena funkcija za optimizacijski problem 2.35 je:

$$f(\mathbf{x}) = \underset{k}{\operatorname{argmax}} [\langle \mathbf{w}_i, \mathbf{x} \rangle + b_i], \quad i = 1, \dots, k.$$

Poglavje 3

Metoda podpornih vektorjev enega razreda

V primeru klasične uporabe metode podpornih vektorjev imamo na voljo učne primere dveh ali več razredov skupaj z ustreznimi oznakami pripadnosti razredu. V nekaterih primerih pa se lahko zgodi, da podatkom niso pripisane ustrezne oznake razredov [4] ali pa imamo na voljo podatke samo enega razreda in je podatke drugega razreda težko določiti [19]. V takih primerih se lahko opremo na uporabo metode podpornih vektorjev enega razreda (ang. *one-class SVM*). Predpostavka metode podpornih vektorjev enega razreda je, da se objekti posameznega razreda grupirajo. Omenjena metoda v splošnem išče hiperravnino, ki ločuje podatke od koordinatnega izhodišča tako, da maksimizira razdaljo med hiperravnino in koordinatnim izhodiščem na eni in podatki (podpornimi vektorji) na drugi strani hiperravnine. Naloga je pod določenimi pogoji enakovredna iskanju hiperkrogle, ki minimizira polmer R in poskuša obenem zajeti čim več podatkov posameznega razreda.

3.1 OSH, ki ločuje koordinatno izhodišče od podatkov učne množice

Naj bo

$$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N) \subset \mathcal{X} \quad (3.1)$$

množica N učnih primerov brez oznake razreda in Φ preslikava iz vhodnega prostora v prostor značilk $\mathcal{X} \rightarrow \mathcal{H}$. Poiskati želimo funkcijo, ki bi zavzemala vrednost $+1$ na področju, ki bi vsebovalo večino učnih primerov, in vrednost -1 na njegovem komplementu. Najprej učne primere preslikamo v prostor značilk in nato poiščemo hiperravnino — OSH, ki poteka med koordinatnim izhodiščem in učno množico (slika 3.1). Optimizacijska naloga se glasi:

$$\underset{\mathbf{w} \in \mathcal{H}, \boldsymbol{\xi} \in \mathbb{R}^M, \rho \in \mathbb{R}}{\text{minimiziraj}} \quad \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu N} \sum_{i=1}^N \xi_i - \rho \quad (3.2)$$

$$\begin{aligned} \text{glede na omejitve} \quad & \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle \geq \rho - \xi_i \\ & \xi_i \geq 0 \quad i = 1, \dots, N \end{aligned} \quad (3.3)$$

S parameterom ν , $0 < \nu \leq 1$, uravnavamo dve nasprotujoči si zahtevi: zahtevo po širokem robu in zahtevo po majhni klasifikacijski napaki.

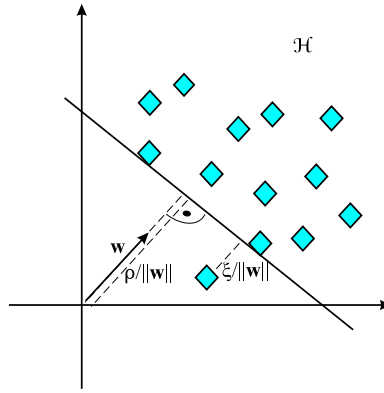
Odločitvena funkcija ima obliko:

$$f(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \Phi(\mathbf{x}) \rangle - \rho) \quad (3.4)$$

oziroma:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^N \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \rho\right), \quad (3.5)$$

če jo zapišemo s pomočjo jedrne funkcije.



Slika 3.1: Metoda podpornih vektorjev enega razreda išče OSH, ki ločuje koordinatno izhodišče in učne primere. Za konkreten ilustrativni primer na sliki leži večina primerov na pravi strani hiperravnine, razen enega, ki pa prispeva neničeln ξ_i . Razdalja med koordinatnim izhodiščem in hiperravnino (odebeljena premica) pri SVM-ju enega razreda ima enak pomen kot razdalja med hiperravnino in mejo pri SVM-ju, ki ločuje med dvema razredoma.

Podobno kot pri SVM-ju, ki ločuje med dvema razredoma, tudi tu uvedemo Lagrangeove spremenljivke $\alpha_i > 0$ in $\beta_i > 0$ ter zapišemo Lagrangeovo funkcijo:

$$f(\mathbf{w}, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu N} \sum_{i=1}^N \xi_i - \rho - \sum_{i=1}^N \alpha_i (\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle - \rho + \xi_i) - \sum_{i=1}^N \beta_i \xi_i. \quad (3.6)$$

Lagrangeovo funkcijo (3.6) odvajamo po \mathbf{w} , ξ in ρ . Po KKT-izreku dobimo naslednje povezave:

$$\mathbf{w} = \sum_{i=1}^N \alpha_i \Phi(\mathbf{x}_i) \quad (3.7)$$

$$\alpha_i = \frac{1}{\nu N} - \beta_i \leq \frac{1}{\nu N} \quad (3.8)$$

$$\sum_{i=1}^N \alpha_i = 1 \quad (3.9)$$

Če vstavimo izraze (3.7-3.9) v Lagrangeovo funkcijo (3.6) in upoštevamo, da je $k(\mathbf{a}, \mathbf{b}) = \langle \Phi(\mathbf{a}), \Phi(\mathbf{b}) \rangle$, dobimo dualni problem oblike:

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^N}{\text{minimiziraj}} \quad \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \quad (3.10)$$

$$\begin{aligned} \text{glede na omejitve} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu N}, \\ & \sum_{i=1}^N \alpha_i = 1 \end{aligned} \quad (3.11)$$

S pomočjo α_i , ki jih dobimo kot rešitev optimizacijskega problema (3.10), in KKT-pogoja (3.8) izračunamo najprej β_i . Kadar sta pri nekem i -ju β_i in α_i različni od nič, velja v skladu s KKT-pogoji: $\xi_i = 0$ in $(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle - \rho + \xi_i) = 0$. Iz tega sledi:

$$\rho = \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle = \sum_{j=i}^N \alpha_j k(\mathbf{x}_j, \mathbf{x}_i) \quad (3.12)$$

Enačbe (3.10-3.12) opisujejo metodo podpornih vektorjev enega razreda z mehкими mejami. Ko pa se ν približuje ničli, se zgornja meja Lagrangeovih multiplikatorjev α_i približuje neskončnosti (3.11). V tem primeru klasifikacijskih napak v učni fazi sploh ne dopuščamo — dobimo problem s strogo mejo. Tudi za problem s strogo mejo vedno obstaja rešitev, saj ρ lahko postane poljubno veliko negativno število omejitve (3.3).

3.2 Hiperkrogla

Za opis učne množice brez oznak \mathcal{X} (3.1) lahko uporabimo tudi hiperkroglo, ki jo konstruiramo tako, da zavzema čim manjši volumen in vsebuje čim večji del učnih primerov. Tudi ta problem lahko formuliramo kot problem s strogo ali mehko mejo. Optimizacijska funkcija ima obliko:

$$\underset{c \in \mathcal{H}, \boldsymbol{\xi} \in \mathbb{R}^M, R \in \mathbb{R}}{\text{minimiziraj}} \quad R^2 + \frac{1}{\nu N} \sum_{i=1}^N \xi_i - \rho \quad \text{pri } 0 < \nu \leq 1 \quad (3.13)$$

$$\begin{aligned} \text{glede na omejitve} \quad & \|\Phi(\mathbf{x}_i) - c\|^2 \leq R^2 + \xi_i \\ & \xi_i \geq 0 \quad i = 1, \dots, N. \end{aligned} \quad (3.14)$$

Problem (3.13) pretvorimo v dualno obliko:

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^M}{\text{minimiziraj}} \quad \sum_{i,j=1}^N \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i k(\mathbf{x}_i, \mathbf{x}). \quad (3.15)$$

$$\begin{aligned} \text{glede na omejitve} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu N} \\ & \sum_{i=1}^N \alpha_i = 1 \quad i = 1, \dots, N. \end{aligned} \quad (3.16)$$

Rešitev c zapišemo:

$$c = \sum_{i=1}^N \alpha_i \Phi(\mathbf{x}_i). \quad (3.17)$$

Odločitvena funkcija pa ima obliko:

$$f(\mathbf{x}) = \text{sgn}\left(R^2 - \sum_{i,j=1}^N \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + 2 \sum_{i=1}^N \alpha_i k(\mathbf{x}_i, \mathbf{x}) - k(\mathbf{x}, \mathbf{x})\right). \quad (3.18)$$

R^2 izračunamo s pomočjo kateregakoli \mathbf{x}_i , pri katerem je $0 < \alpha_i < \frac{1}{\nu N}$. Za vse take \mathbf{x}_i -je namreč velja, da je vrednost sgn funkcije (3.18) enaka 0.

Za jedra $k(\mathbf{x}, \mathbf{x}')$, kjer nastopa neodvisna spremenljivka v obliki $\mathbf{x} - \mathbf{x}'$, npr. RBF-jedro, velja, da je $k(\mathbf{x}, \mathbf{x})$ konstanta. Za taka jedra velja, da sta optimizacijska problema (3.10, 3.11) in (3.15, 3.16) ekvivalentna. Isto velja tudi za odločitveni funkciji (3.5) in (3.18).

Poglavje 4

Metoda BW

Dokler imamo pri razpoznavanju objektov v učni in testni množici enako ozadje, metoda SVM deluje dobro. Z variiranjem ozadja pa se točnost SVM-klasifikatorja zmanjša. Na klasifikacijo oz. položaj točke v prostoru namreč vplivajo tako slikovni elementi, ki pripadajo ozadju, kakor tudi slikovni elementi, ki pripadajo predmetu. Različni prizori drastično vplivajo na pravilnost SVM-klasifikatorja. Ena izmed metod, kako zaobiti problem različnega ozadja, je metoda BW (ang. *Black-white method*, BW).

4.1 Ideja BW-metode

Metoda BW spada v skupino metod pedagoškega učenja. Ideja metode BW izhaja iz spoznanja, da se raznolikega ozadja lahko naučimo in s tem povečamo neobčutljivost klasifikatorja SVM na spreminjajoče se ozadje. Recimo, da imamo N slik dimenije M . V vsaki sliki M_{bg} slikovnih elementov pripada ozadju; ($M_{bg} < M$). Vrednosti teh slikovnih elementov na klasifikacijo ne smejo vplivati, zato poskušamo njihov vpliv čim bolj zmanjšati.

Zgradimo novo učno množico tako, da slikovne elemente ozadja zamenjujemo s poljubnimi vrednostmi. Kot dodatne primere za učno množico bi torej lahko generirali vse možne slike ozadja. Tako generirana množica bi bila v obsegu realnih števil neskončno velika in v praksi tehnično neobvladljiva. Izredno bi se povečala tako zahteva po spominu kakor tudi čas, potreben za učenje.

Druga možnost je, da naključno generiramo samo podmnožico slik z različnim ozadjem.

V tem primeru je težko napovedati, koliko naključno generiranih slik zadostuje, da bi se sposobnost generalizacije klasifikatorja zadostno povečala.

Roobaert [18] je predlagal metodo učenja z množico slik predmetov na ozadju z ekstremnima vrednostma.

Roobaert predpostavi, da je velikost učne množice manjša ali pa reda velikosti dimenzije prostora ($N < M$). Ta predpostavka v večini primerov zagotavlja, da sta množici za poljubna dva razreda linearno ločljivi, kar pomeni, da je linearni klasifikator dovolj dobro orodje za klasifikacijo le-teh.

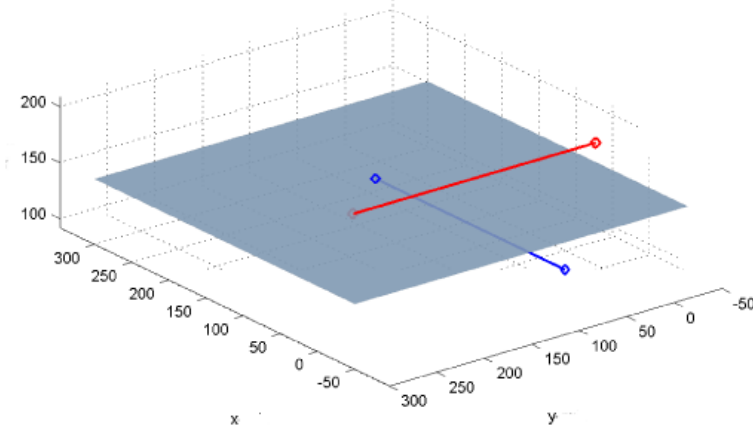
4.1.1 Delež slikovnih elementov, ki pripadajo ozadju, je majhen ($M_{bg} \ll M$)

Recimo, da uporabljamo linearni klasifikator in da so vrednosti podatkov omejene. Pod takimi pogoji je dovolj, da razširjeno učno množico S_0 razširimo v S_1 tako, da zamenjamo vseh M_{bg} slikovnih elementov z vsemi možnimi kombinacijami ekstremnih vrednosti. V primeru slik so vrednosti podatkovne matrike navadno omejene na interval $[0, 255]$. Ekstremni vrednosti sta torej 0 in 255. Velikost S_1 je $N \cdot 2^{M_{bg}}$. Pod pogojem, da je S_1 še vedno linearno ločljiva množica, je OSH vzporedna z osmi, ki predstavljajo ozadje. Pogoj linearne ločljivosti je za S_1 še vedno izpolnjen, če velja $N \cdot 2^{M_{bg}} \leq (d + 1)$. Za majhno število slikovnih elementov, ki pripadajo ozadju M_{bg} , opisana metoda, **Metoda 1**, omogoča popolno neobčutljivost na slikovne elemente ozadja.

Opisano metodo bomo ilustrirali na 3-dimenzionalnem primeru. Naj bosta a spodnja in b zgornja ekstremna vrednost. Imamo dva razreda, v katera bi radi klasificirali podatke, imenujmo ju pozitivni in negativni razred. Predstavniki negativnega razreda imajo obliko $(*, y_-, z_-)$, predstavniki pozitivnega pa $(x_+, *, z_+)$, $z_- \neq z_+$. Simbol $*$ označuje slikovne elemente ozadja. Vse možne predstavnike posameznega razreda lahko v 3-D prostoru predstavimo z daljico. Če bi vzeli po en naključni primer iz vsakega razreda in izračunali OSH, bi ta z veliko verjetnostjo sekala kakšno izmed dveh daljic, kar pa pomeni napačno klasifikacijo. Če pa učno množico S_1 (4.1) generiramo po Metodi 1 z vsemi možnimi kombinacijami vrednosti na točkah ozadja, dobimo OSH, ki nikoli ne seka nobene izmed daljic. Enačba OSH-ja se v tem primeru glasi: $z = \frac{z_- + z_+}{2}$. Taka hiperravnina nam zagotavlja

pravilno klasifikacijo ne gleda na vrednosti, ki jih imajo slikovni elementi ozadja. Primer ilustrira slika 4.1.

$$S_1 = \{(a, y, z_-), (b, y, z_-), (x_+, a, z_-), (x_+, b, z_-)\} \quad (4.1)$$



Slika 4.1: Ilustracija Metode 1 za $M = 3$ in $M_{bg} = 1$. Učno množico predstavljajo modri in rdeči krogi. Modra in rdeča daljica vključujeta primere vseh možnih ozadij. OSH (modra ravnina) omogoča klasifikacijo, ki je neobčutljiva na vrednost slikovnega elementa, ki pripada ozadju. Opazimo lahko tudi, da je OSH vzporedna z osema, ki predstavljata ozadje v prvi in drugi učni množici (x-os in y-os).

4.1.2 Delež slikovnih elementov, ki pripadajo ozadju, je relativno velik — aproksimacijska metoda

Delež slikovnih elementov, ki pripadajo ozadju, je navadno relativno velik. M_{bg} je le nekoliko manjši od M . Glede na Metodo 1 velikost množice S_1 raste eksponentialno z M_{bg} . Metoda 1 je primerna le za majhne vrednosti M_{bg} , zato Roobaert [18] predlaga dve aproksimacijski metodi:

Metoda 2: Uporabimo naključno podmnožico množice S_1 .

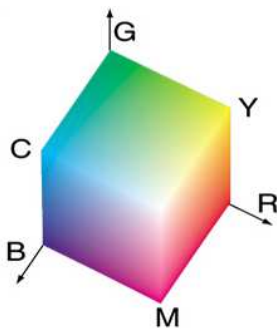
Metoda 3: Uporabimo samo ekstremne pare iz S_1 . V tem primeru se učna množica podvoji. V primeru slik to pomeni, da so objekti postavljeni na belo in črno ozadje.

To metodo imenujemo BW-metodo (ang. *black white method*) in se izkaže kot boljša aproksimacijska različica.

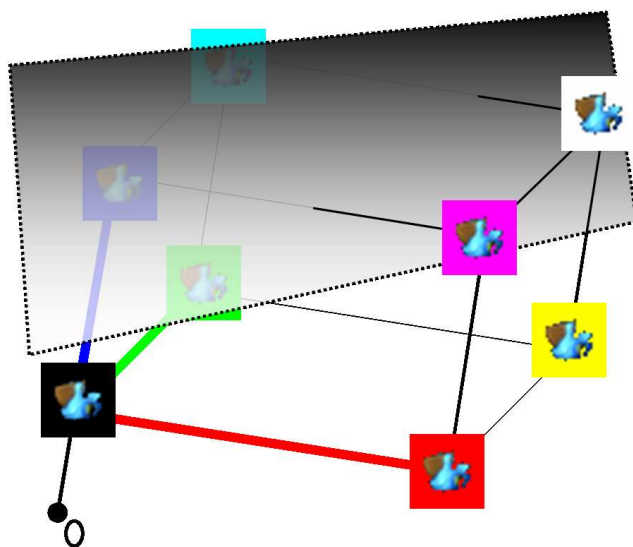
4.2 Barvna BW-metoda

V primeru sivinskih slik si grafično res lahko predstavljamo dve ekstremni vrednosti: črno in belo. V primeru barvnih (RGB) slik pa je ekstremnih vrednosti dejansko več. Spreminjanje barv v RGB prostoru si lahko predstavljamo kot potovanje po kocki (slika 4.2).

Sliko, ki predstavlja točko v hiperprostoru, predstavimo z dvema vektorjema. Prvi vektor predstavlja premik iz izhodišča, ki ga določajo slikovni elementi, ki pripadajo objektu, drugi vektor pa premik, ki ga določajo slikovni elementi, ki pripadajo ozadju. V primeru, da je vektor, ki predstavlja premik zaradi spremembe ozadja, velik v primerjavi z vektorjem, ki predstavlja slikovne elemente objekta, se kaj hitro zgodi, da hiperavrnina, generirana na učni množici s črnim in z belim ozadjem, objekte na nekaterih ozadjih napačno klasificira (slika 4.3). Zato je v primeru barvnih slik smiselno povečati učno množico še z nekaterimi slikami, kjer ozadje zamenjamo še z drugimi ekstremnimi RGB vrednostmi, ki so poleg bele (255, 255, 255) in črne (0, 0, 0) še rdeča (255, 0, 0), škrlatna (255, 0, 255), modra (0, 0, 255), turkizna (0, 255, 255), zelena (0, 255, 0) in rumena (255, 255, 0) barva.



Slika 4.2: Barvni prostor RGB



Slika 4.3: Barvna BW-metoda. Če se v primeru RGB slik raznolikosti ozadja učimo le na belem in črnem ozadju, se lahko zgodi, da OSH napačno klasificira kar velik del ozadij.

Poglavje 5

Metoda glavnih komponent

Slike so predstavljene z visokodimenzionalnimi podatki. Vsebina vektorjev so kar vrednosti s katerimi so predstavljeni posamezni slikovni elementi. Ker je delo z velikimi podatkovnimi strukturami tako časovno kot tudi prostorsko zahtevno je priročno, če nam uspe preslikati visokodimenzijske vektorje značilk v nižje dimenzijske in pri tem ohraniti vso potrebno informacijo. Priljubljene so linearne preslikave, saj so lahko izračunljive in analitično obvladljive [6]. V računalniškem vidu se v zadnjih letih pogosto uporablja *metoda glavnih komponent* (ang. *Principal Component Analyses, PCA*), ki išče projekcijo, ki najbolje predstavi podatke v smislu napake najmanjših kvadratov. Metoda glavnih komponent je v primeru razpoznavanja objektov na poljubnem ozadju zanimiva zato, ker obstaja način, ki omogoča njeno uporabo na podmnožici značilk, kar vodi k določeni stopnji neobčutljivosti na šum.

5.1 Osnovni PCA

Ideja PCA-ja je, da poiščemo smeri v visokodimenzijskem prostoru značilk, ki imajo največjo varianco pri dani učni množici. Smeri, v kateri je varianca manjša, navadno modelirajo šum. Te smeri lahko enostavno zavržemo in tako zmanjšamo dimenzijo prostora. [13]

Najprej izračunamo M -dimenzionalni povprečni vektor μ in kovariančno matriko C .

$$\boldsymbol{\mu} = [\mu_1, \dots, \mu_M] = \frac{1}{N} \sum_{j=1}^N \mathbf{x}_j \quad (5.1)$$

$$\mathbf{C} = \frac{1}{N} \hat{\mathbf{X}} \hat{\mathbf{X}}^T, \quad (5.2)$$

kjer $\hat{\mathbf{X}}$ v našem primeru predstavlja slike $\mathbf{x}_1, \dots, \mathbf{x}_N$, zložene po stolpcih, od katerih smo odšteli povprečje $\boldsymbol{\mu}$:

$$\hat{\mathbf{X}} = [\mathbf{x}_1 - \boldsymbol{\mu}, \dots, \mathbf{x}_N - \boldsymbol{\mu}] \quad (5.3)$$

Če matriko \mathbf{C} diagonaliziramo, dobimo ortonormalno matriko \mathbf{U} in diagonalno matriko Λ :

$$\mathbf{C} = \mathbf{U} \Lambda \mathbf{U}^T. \quad (5.4)$$

Ortonormalna matrika $\mathbf{U} = [\mathbf{e}_1, \dots, \mathbf{e}_N] \in \mathbb{R}^{M \times N}$ v stolpcih vsebuje lastne vektorje $\mathbf{e}_1, \dots, \mathbf{e}_N$. Elementi na diagonalni matrike $\Lambda \in \mathbb{R}^{N \times N}$ so lastne vrednosti $\lambda_1, \dots, \lambda_N$. Predpostavljamo, da so lastne vrednosti in ustrezni lastni vektorji urejeni po padajočem vrstnem redu glede na velikost lastne vrednosti. Največja varianca podatkov je v smeri lastnih vektorjev, ki ustrezajo največji lastni vrednosti. Navadno za opis podatkov zadošča prvih nekaj lastnih vektorjev $\mathbf{e}_1, \dots, \mathbf{e}_p$, kjer je $p \ll N$. Prostor, ki ga razpenja prvih p lastnih vektorjev imenujemo lastni prostor. Preslikavo slike \mathbf{x} v p -dimenzionalni prostor zapišemo:

$$\mathbf{a} = \mathbf{A}^T \hat{\mathbf{x}}, \quad (5.5)$$

kjer so stolpci matrike \mathbf{A} sestavljeni iz prvih p lastnih vektorjev. $\hat{\mathbf{x}}$ pa je $\mathbf{x} - \boldsymbol{\mu}$.

Rekonstrukcijo slike $\hat{\mathbf{x}}$ iz p -dimezionalnega prostora izrazimo:

$$\tilde{\mathbf{x}} = \sum_{i=1}^p a_i(\mathbf{x}) \mathbf{e}_i \quad (5.6)$$

5.2 Robustni PCA

Pomanjkljivost osnovne metode glavnih komponent je, da metoda lahko odpove, že če spremenimo en slikovni element slike \mathbf{x} . Predpostavljamo, da je j -ta komponenta vektorja slike \mathbf{x} spremenjena za δ , kar pomeni, da so komponente $a_i, i = 1, \dots, p$ spremenjene:

$$\tilde{a}_i = a_i + \delta e_{i_j} \quad (5.7)$$

Razlika $\|\tilde{\mathbf{a}} - \mathbf{a}\|$ je lahko poljubno velika, kar pomeni, da je metoda nerobustna, točka zloma je 0%. Leonardis in soavtorji [10] so predlagali robustno metodo, ki se od klasične razlikuje v načinu izračunavanja koeficientov a_i . Medtem ko pri standardni metodi izračunamo koeficiente s projekcijo podatkov na lastne vektorje, robustna metoda temelji na iskanju ustreznih vrednosti koeficientov na podlagi podmnožic slikovnih elementov testne slike. Sestavljata jo dva bistvena koraka: generiranje hipotez in izbira najboljše hipoteze.

5.2.1 Generiranje hipotez

Predpostavimo, da bi za predstavitev podatkov vzeli vse izračunane lastne vektorje $k = N$ in da slike $\mathbf{x}_i, i = 1, \dots, N$, ne vsebujejo nobenega šuma. Koeficiente \mathbf{a}_i (enačba 5.5) lahko v tem primeru izračunamo s sistemom N linearnih enačb. Vzamemo podmnožico, sestavljeno iz N slikovnih elementov $\mathbf{r} = (r_1, r_2, \dots, r_N)$ in zapišemo sistem linearnih enačb:

$$x_{r_i} = \sum_{j=1}^N a_j(\mathbf{x}) e_{j_{r_i}}, \quad 1 \leq i \leq N \quad (5.8)$$

Če pa sliko, ki lahko vsebuje tudi šum, zapišemo kot približek linearne kombinacije p lastnih slik, pri čemer velja $p < N$, sistemu enačb (5.8) ne moremo zadostiti. Namesto sistema 5.8 nastavimo sistem enačb, ki ima več enačb kot spremenljivk — predeterminiran sistem zgrajen iz k , $\mathbf{r} = (r_1, r_2, \dots, r_k)$, točk ($p < k \leq N$). Število slikovnih elementov je v splošnem veliko večje kot število lastnih slik. Iščemo vektor \mathbf{a} , ki v smislu najmanjših kvadratov minimizira spodnji sistem enačb:

$$E(\mathbf{r}) = \sum_{i=1}^k \left(x_{r_i} - \sum_{j=1}^p a_j(\mathbf{x}) e_{j_{r_i}} \right)^2 \quad (5.9)$$

Generiranje posamezne hipoteze navadno poteka v več iteracijah. Testno sliko poskušamo aproksimirati z izbranim naborom lastnih slik. Začnemo z naključno podmnožico \mathcal{K} , ki vsebuje k slikovnih elementov iz testne slike, nastavimo sistem enačb (5.9) in izračunamo vektor $\mathbf{a} \in \mathbb{R}^p$. Na podlagi distribucije rekonstrukcijske napake zmanjšamo podmnožico \mathcal{K} , tako da odstranimo točke z največjo napako (ang. *α -trimming*). Postopek ponavljamo dokler število slikovnih elementov v množici \mathcal{K} ne doseže minimalnega dovoljenega števila.

Na podlagi različnih podmnožic slikovnih elementov generiramo več hipotez. Seveda vse ne ustrezajo dobremu opisu testne slike. Vendar pa večje kot je število hipotez večja je možnost, da so med njimi tudi take, ki smo jih izračunali na nešumnih podatkih. V koraku izbire poskušamo poiskati tisto, ki testno sliko kjub šumu najbolj opiše. Ko enkrat izračunamo PCA-koeficiente testne slike, le to lahko klasificiramo. Za klasifikacijo lahko uporabimo različne metode: metodo najbližjega soseda (ang. *nearest neighbor*, *NN*), metodo podpornih vektorjev idr.

5.2.2 Izbira hipoteze

Pri izbiri najboljše hipoteze se lahko poslužimo več kriterijev. Končna izbira kriterijev je precej odvisna od narave problema. V nadaljevanju bomo opisali dva načina izbire najboljše hipoteze: na podlagi principa najkrajše dolžine opisa (ang. *Minimum Description Length*, *MDL*) in na podlagi glasovanja glede na oddaljenost hipoteze od testnega primera. Pri tem bomo predpostavljali, da se na sliki nahaja le en znan objekt.

Način izbire hipoteze lahko razširimo tudi tako, da dopuščamo, da je na eni sami sliki več objektov [10].

Izbira hipoteze po principu najkrajše dolžine opisa

Vsako izmed generiranih hipotez \mathbf{a}_i za testno sliko \mathbf{x} ocenimo s pomočjo MDL-kriterija [10]. Tisto hipotezo, ki ima največjo vrednost MDL-kriterijske funkcije izberemo kot najboljšo in jo uporabimo za klasifikacijo.

Za vse generirane hipoteze \mathbf{a}_i , ki pripadajo testni sliki \mathbf{x} , izračunamo rekonstrukcijo $\tilde{\mathbf{x}}_i$ (enačba 5.6) in vektor napake $\boldsymbol{\xi}_i$:

$$\boldsymbol{\xi}_i = (\mathbf{x} - \tilde{\mathbf{x}}_i)^T (\mathbf{x} - \tilde{\mathbf{x}}_i). \quad (5.10)$$

Nato poiščemo množico kompatibilnih točk $D_i = \{j | \xi_j^2 < \Theta\}$. S s_{i1} označimo število kompatibilnih točk, $s_{i1} = |D|$, s s_{i2} pa vsoto napak na teh točkah, $s_{i2} = \sum_{D_{ij}} \xi_j$. Kriterijsko funkcijo zapišemo:

$$F(i) = K_1 \cdot s_{i1} - K_2 \cdot s_{i2} - K_3 \cdot p_i, \quad (5.11)$$

kjer koeficiente K_1 , K_2 in K_3 določimo bodisi eksperimentalno bodisi avtomatsko [11].

Izbira hipoteze na podlagi glasovanja

Znano je, [10], da so generirane hipoteze, ki so bližje učnim primerom, boljše kot tiste, ki so od učnih primerov bolj oddaljene. To dejstvo je osnova za t.i. glasovalni algoritem (ang. *voting algorithm*), ki slike klasificira na podlagi oddaljenosti generiranih hipotez od učnih primerov [1].

Ko enkrat izračunamo hipoteze na podlagi robustnega PCA pristopa, za vsako hipotezo izberemo najbližjo točko iz mnogoterosti. Tako izbrana točka dobi glas, katerega vrednost je obratno sorazmerna velikosti razdalje med robustno generiranimi PCA koeficienti in izbrano točko na mnogoterosti ¹. Vrednost glasovalne funkcije $v(d)$ (5.12) je omejena na intervalu $[0, 1]$.

$$v(d_{ij}) = e^{\frac{-d_{ij}^2}{2\sigma^2}}, \quad (5.12)$$

kjer je

¹Točke \mathbf{a}_i , ki pripadajo objektu tvorijo hiperploskev v p dimenzionalnem prostoru. Mnogoterost je pojem, ki obsega hiperploskve vseh učnih primerov.

$$d_{ij} = \arccos \frac{\mathbf{a}_i^T \mathbf{a}_j}{\|\mathbf{a}_i\| \|\mathbf{a}_j\|} \quad (5.13)$$

kot med vektorjem \mathbf{a}_i , ki ga izračunamo z robustnim postopkom in vektorjem \mathbf{a}_j , ki pripada najbližji točki iz mnogoterosti. Sliko klasificiramo kot učni primer, ki je zbral največji seštevek glasov.

Poglavje 6

Hierarhični SVM

6.1 Definicija osnovnih pojmov

6.1.1 Maska

Vsaki sliki \mathbf{x}_i , $i = 1, \dots, N$ iz učne množice priredimo sliko B_i , ki kodira pripadnost slikovnih elementov k objektu ali ozadju. Binarne slike imenujemo slike B_i ; $i = 1, \dots, N$, ki imajo vrednost 1, kjer slikovni element predstavlja objekt, in vrednost 0, kjer je na sliki ozadje.

$$B_i(m, n) = \begin{cases} 1 & \text{če je } x_i(m, n) \text{ slikovni element, ki pripada objektu} \\ 0 & \text{če je } x_i(m, n) \text{ slikovni element, ki pripada ozadju,} \end{cases} \quad (6.1)$$

$n = 1, \dots, c$ in $m = 1, \dots, l$, kjer je c število stolpcev, l pa število vrstic slikovne matrike.

6.1.2 Maska preseka

Naj slike \mathbf{x}_j in njihove maske B_j , $j = 1, \dots, k$ pripadajo le enemu objektu. Če nad binarnimi slikami B_j izvajamo AND operacijo, dobimo masko preseka M^p . Maska preseka ima vrednost 1 na tistih mestih v slikovni matriki, kjer istoležni slikovni elementi na vseh

slikah x_j predstavljajo objekt, in vrednost 0 sicer.

$$M^p(m, n) = \bigwedge_{j=1}^k B_j(m, n) \quad m = 1, \dots, l, \quad n = 1, \dots, c. \quad (6.2)$$

6.1.3 Maska unije

Če nad slikami B_j , $j = 1 \dots k$, izvajamo OR operacijo, dobimo masko unije M^u . Maska unije ima vrednost 1 na tistih mestih v slikovni matriki, kjer istoležni slikovni elementi na vsaj eni izmed slik x_j predstavljajo objekt, in vrednost 0 v primeru, da ustrezen slikovni element na vseh slikah predstavlja ozadje.

$$M^u(m, n) = \bigvee_{j=1}^k B_j(m, n) \quad m = 1, \dots, l, \quad n = 1, \dots, c. \quad (6.3)$$

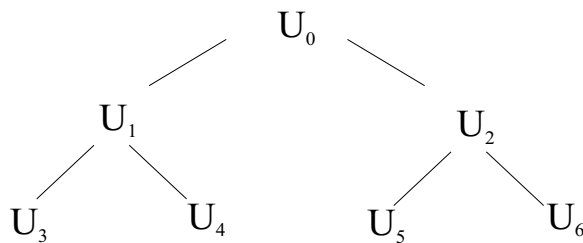
6.1.4 Velikost maske

Velikost maske definiramo kot število takih slikovnih elementov, ki so v maski enaki 1.

$$v = \sum_{m=1}^c \sum_{n=1}^l M_{m,n} \quad (6.4)$$

6.2 Hierarhična ureditev

Učno množico za vsak posamezni učni predmet uredimo v hierarhično strukturo, ki ima obliko 3-nivojskega binarnega drevesa. Na najvišjem, to je na prvem nivoju, imamo celo učno množico U_0 . Na drugem nivoju učno množico razdelimo na dva dela, U_1 in U_2 , in na tretjem vsako podmnožico še nadalje razdelimo na dva dela. Na tretjem nivoju dobimo tako 4 podmnožice U_3 , U_4 , U_5 ter U_6 (slika 6.1). Način delitve je opisan v nadaljevanju.



Slika 6.1: Hierarhična delitev učne množice.

6.2.1 Delitev množice

Enostavna delitev množice

Pri enostavni delitvi zložimo slike po vrsti glede na rotacijski kot. Delitev, ki upošteva rotacijski kot, je smiselna, ker so si slike z manjšo razliko v zornem kotu praviloma bolj podobne.

Množico prvega nivoja predstavljajo vse učne slike. Množici drugega nivoja dobimo tako, da enostavno razpolovimo množico iz prvega nivoja. V levi veji, vozlišče 1, imamo tako prvo polovico učnih slik, v desni veji poddrevesa, vozlišče 2, pa drugo polovico. Tudi naslednje nivoje dobimo na enak način, z razpolavljanjem. Tako vozlišče 3 vsebuje prvo četrtino vseh učnih slik oz. prvo polovico učnih slik vozlišča 1 itd.

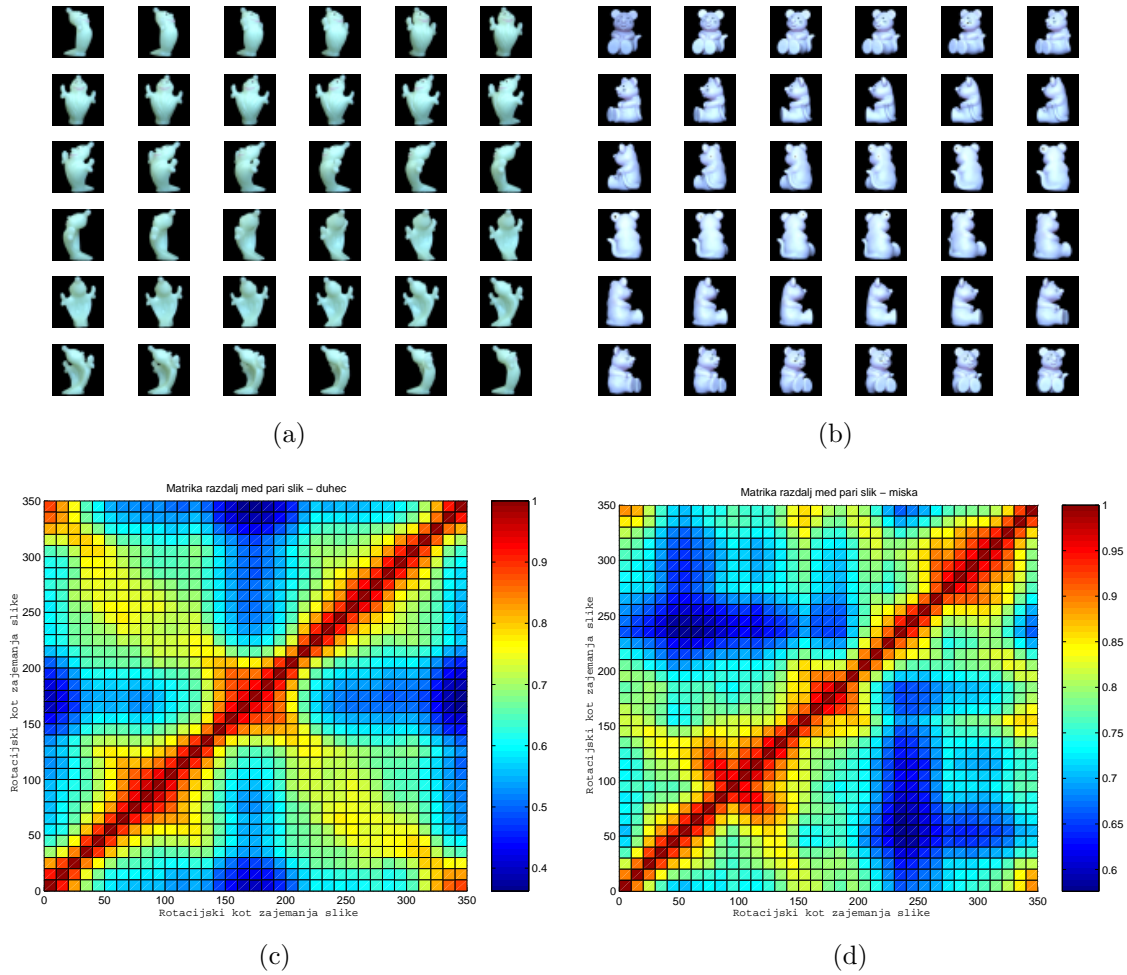
Pri delitvi učne množice si želimo, da bi si bile slike posamezne podmnožice med seboj čim bolj podobne in bi tvorile čim večjo masko presekov. Enostavna delitev tega ne zagotavlja. Delitev lahko izboljšamo z upoštevanjem razmerja med unijo in presekom, ki ju izračunamo med dvema slikama B_i , B_{i+1} , ki sta si sosednji glede na kot zajemanja slik.

Delitev z upoštevanjem velikosti mask v primeru enorazrednih SVM-jev

Tudi v tem primeru slike uredimo glede na kot pogleda. Iščemo tako delitev učne množice \mathcal{S} na dva dela \mathcal{S}_1 in \mathcal{S}_2 , da bo velikost obeh mask presekov čim večja in da bo v vsaki izmed podmnožic U_1 , U_2 vsaj neko minimalno število primerkov.

Uvedemo mero oz. stopnjo podobnosti m med dvema maskama B_i in B_j , $i, j = 1, \dots, N$, ki jo definiramo kot razmerje med velikostjo maske preseka $v_p(B_i, B_j)$ in ve-

likostjo maske unije $v_u(B_i, B_j)$.



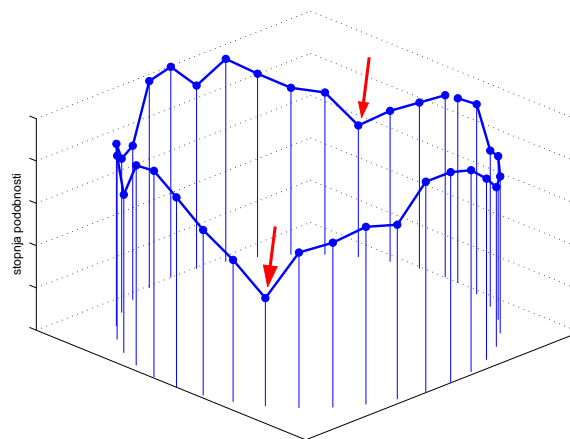
Slika 6.2: Merjenje podobnosti mask. V zgornji vrstici so prikazane slike dveh objektov, za katere izračunamo mero podobnosti. Slike predmetov so zložene po vrsti glede na rotacijski kot zajemanja. Slika (c) prikazuje podobnost med maskami, ki pripadajo objektu, prikazanemu pod (a), slika (d) pa podobnost med maskami, ki pripadajo objektu pod (b). Slikovni element (i, j) v matriki meri podobnost med maskama B_i in B_j . Vidimo, da je podobnost mask med slikama, ki imata podoben zorni kot, večja kot med maskama, ki pripadata slikama z različnim zornim kotom.

stopnja podobnosti = velikost preseka / velikost unije

$$m_{i,j} = m(B_i, B_j) = \frac{v_p(B_i, B_j)}{v_u(B_i, B_j)} \quad (6.5)$$

Za vse pare mask sosednjih slik izračunamo velikost maske preseka in unije ter stopnjo podobnosti. Iz slike 6.2 je razvidno, da je podobnost med dvema maskama istega predmeta, ki sta sosednji glede na rotacijski kot, v splošnem večja kot med dvema naključno izbranimi maskama istega predmeta. Smiselno je torej urediti maske glede na kot pogleda in računati stopnje podobnosti le med sosednjima maskama.

Shematski prikaz mer za pare sosednjih slik v primeru čajnika je na sliki 6.3.



(a) Mera podobnosti



(b) Slike čajnika

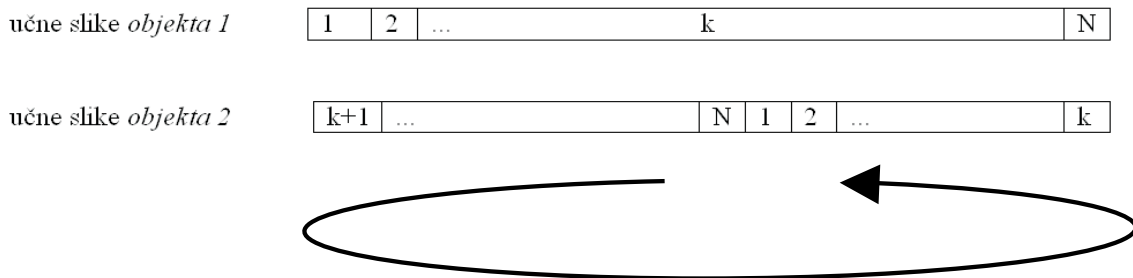
Slika 6.3: Mere podobnosti (a) za slike čajnika (b), ki so si sosednje glede na kot zajemanja. Razvrščene so po krogu, ki nakazuje smer pogleda na objekt. Rdeči puščici nakazujeta mesti delitve.

Za delitev množice na dva dela (slika 6.3) potrebujemo dve presečni mesti. Delitev je smiselna tam, kjer se maski med seboj najbolj razlikujeta. Sosednji par slik z najmanjšo mero podobnosti predstavlja prvo presečno mesto. Drugo presečno mesto pa dobimo tako, da poiščemo naslednjo najmanjšo mero podobnosti. Če je to mesto tako, da sta po deljenju množice \mathcal{S} obe podmnožici \mathcal{S}_1 in \mathcal{S}_2 dovolj veliki, množici razdelimo. V nasprotnem primeru poiščemo naslednjo najmanjšo *mero podobnosti* in ponovno poskušamo deliti. Postopek ponavljamo toliko časa, dokler nimata obe podmnožici zadovoljive velikosti.

Delitev z upoštevanjem velikosti mask v primeru dvorazrednih SVM-jev

V primeru dvorazrednih SVM-jev moramo upoštevati učni množici dveh razredov. Ena izmed učnih množic najprej zamaknemo tako, da je vsota *stopenj podobnosti* med istoležnimi slikami največja, pri čemer še vedno ohranimo vrstni red slik glede na zorni kot (enačba 6.6, slika 6.4).

$$\underset{p}{\text{maksimiziraj}} \sum_{i=1}^N m_{i,(i+p) \bmod(N)} \quad p = 0, \dots, N-1 \quad (6.6)$$



Slika 6.4: Poravnava učnih množic v primeru binarnega SVM-ja

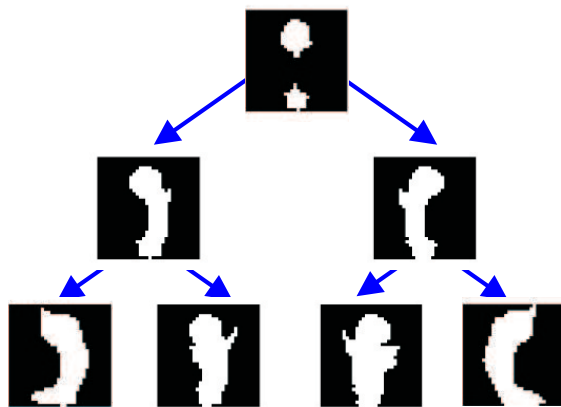
Zaradi enostavnejše notacije zamaknjene slike ponovno oštevilčimo od $1, \dots, N$ tako, da imajo istoležne slike isti indeks (slika 6.5). Presečna mesta določimo na enak način kot pri enorazrednem SVM-ju, le da tu stopnjo podobnosti izračunamo na podlagi štirih slik (slika 6.6), in sicer dveh zaporednih iz učne množice prvega objekta ($\mathbf{x}_i^{obj1}, \mathbf{x}_{(i+1) \bmod(N)}^{obj1}$)

6.3 Hierarhija SVM-jev enega razreda

6.3.1 Gradnja hierarhije

Po hierarhični delitvi učne množice na podmnožice U_0, U_1, \dots, U_6 izračunamo za vsako podmnožico ustrezno masko preseka. Dobimo drevesno strukturo sedmih mask $M_0^P, M_1^P, \dots, M_6^P$ (Slika 6.7). Beli del maske na spodnjih nivojih vedno vsebuje beli del mask na zgornjih nivojih. V fazi učenja vse slike iz posameznih podmnožic U_0, U_1, \dots, U_6 pomnožimo z masko, ki pripada posamezni podmnožici. Za tako preoblikovane slike izračunamo ustrezne hiperkrogle po enačbi 3.15.

Nekaj primerov tako pridobljenih učnih slik je na sliki 6.8.



Slika 6.7: Hierarhija mask za primer slik duhca

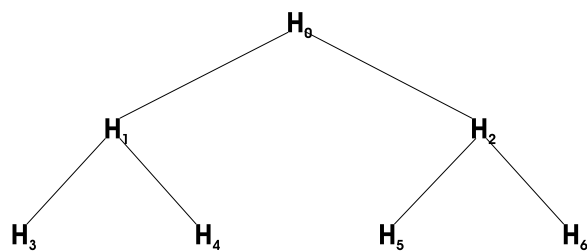
Za vsako izmed sedmih učnih podmnožic izračunamo enorazredni SVM - hiperkroglo (enačba 3.15). Dobimo hierarhijo sedmih hiperkrogel: H_0, H_1, \dots, H_6 (slika 6.9).

6.3.2 Faza razpoznavanja

Recimo, da ugotavljamo, ali se na testni sliki nahaja *objekt 1*. Najprej testno sliko prekrijemo z masko K_0 , ki pripada učnim slikam *objekta 1*. Nato izračunamo odločitveno funkcijo za ustrezno hiperkroglo H_0 (enačba 3.18). Če ima funkcija vrednost ≥ 0 , kar

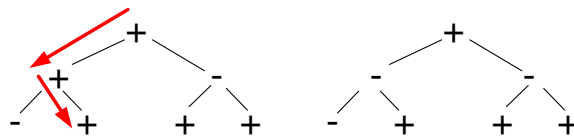
(a) Učne slike maskirane z masko K_0 (b) Učne slike maskirane z masko K_1

Slika 6.8: Nekaj primerov učnih slik z masko



Slika 6.9: Hierarhija hiperkrogel

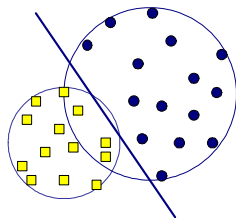
pomeni, da maskirani del testne slike leži znotraj hiperkrogle H_0 , potem je možno, da slika prikazuje *objekt 1*. Razpoznavanje v tem primeru lahko nadaljujemo na drugem nivoju, kjer testno sliko zopet prekrijemo z maskama 2. nivoja in izračunamo ustrezni odločitveni funkciji. V primeru uspešnega testa na drugem nivoju nadaljujemo testiranje na tretjem nivoju. Objekt na sliki razpoznamo, če obstaja pot v hierarhiji od korena pa do listov, ki da skladno odločitev - vedno glasuje za *objekt 1* (slika 6.10).



Slika 6.10: Primer klasifikacije za hierarhijo enorazrednega SVM-ja. V levem primeru smo na sliki razpoznali objekt, saj obstaja dosledna potrditev testa na poti od korena pa do listov drevesne hierarhije. V desnem drevesu pa ni doslednih potrditev testa na poti in zato tu razpoznavanje ni potrjeno.

6.4 Ločitev prekrivajočih se krogel z dvorazrednim SVM-jem

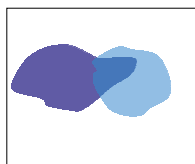
Lahko se zgodi, da se hiperkrogle posameznih razredov prekrivajo. Posledica je, da je posamezna slika lahko klasificirana na dva načina, npr. kot *objekt 1* in *objekt 2*. V takih primerih med podatki dveh hiperkrogel poiščemo še OSH (slika 6.11), ki potrdi ali zavrže razpoznavo.



Slika 6.11: Kombinacija eno- in dvorazrednih SVM-jev

6.4.1 Faza gradnje dvorazrednih SVM-jev

Za vse pare objektov preverimo, če pripadajo skupini *kritičnih parov*. Kritični pari so posledica “skupnih” področij, ki ležijo na preseku mask. Hiperkrogli se lahko v tem primeru sekata in ne določata enolične klasifikacije. V tem primeru določimo mejo med množicama še z dvorazrednim SVM-jem. Slikovni elementi preseka so pomembni zato, ker so v nekaterih situacijah edini del na sliki, ki lahko ločuje med obema objektoma. Recimo, da je na sliki objekt, ki določa temno modro masko (slika 6.12). Objekta, ki določa svetlo modro masko na sliki sicer ni, vendar ga enorazredni SVM-ji zaradi podobnosti ozadja z objektom na sliki najdejo. Kadar vemo, da na sliki ni dveh objektov, se za enega lahko odločimo le na podlagi slikovnih elementov preseka obeh mask oz. hiperravnine, ki ločuje množici obeh objektov na podlagi slikovnih elementov v preseku.



Slika 6.12: Presek dveh mask

Postopek preverjanja pripadnosti *kritičnim parom* je opisan v naslednjem razdelku.

Preverjanje pripadnosti kritičnim parom

Recimo, da za par *objekt 1* in *objekt 2* preverjamo, če pripada t.i. *kritičnim parom*. Masko prvega nivoja za *objekt 1* označimo s K_0^{obj1} , masko prvega nivoja *objekta 2* pa s K_0^{obj2} . Izračunamo novo masko $K_0^{obj1 \& obj2}$, ki je presek maske K_0^{obj1} in K_0^{obj2} :

$$K_0^{obj1 \& obj2} = K_0^{obj1} \bigwedge K_0^{obj2}.$$

Nato z masko $K_0^{obj1 \& obj2}$ pomnožimo učno množico U_0^{obj1} ter za tako pridobljene podatke izračunamo hiperkroglo H_0^{obj1} . Na isti način izračunamo še hiperkroglo za *objekt 2* H_0^{obj2} . Hiperkrogli se sekata, če je razdalja med središči c_1 in c_2 hiperkrogel H_0^{obj1} in H_0^{obj2} manjša kot vsota radijev obeh hiperkrogel R_1 in R_2 :

$$d(c_1, c_2) < R_1 + R_2. \quad (6.8)$$

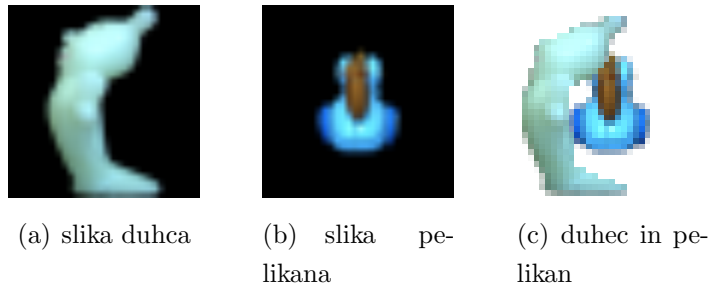
Kadar se ti dve hiperkrogli sekata, par (*objekt 1*, *objekt 2*) pripada skupini kritičnih parov.

Če bi bila narava podatkov taka, da bi se podatki v veliki meri pojavljali na robu hiperkrogle, bi bilo smiselno omejitev 6.8 nekoliko zaostriti tako, da radij hiperkrogel nekoliko povečamo s parametrom t :

$$d(c_1, c_2) < R_1 + R_2 + t, \quad (6.9)$$

ki ga lahko določimo eksperimentalno.

Če je velikost maske $K_0^{obj1 \& obj2}$ zelo majhna (< 15 slikovnih elementov), potem hiperkrogel H_0^{obj1} in H_0^{obj2} sploh ne računamo, ampak privzamemo, da se ne sekata. V takem primeru lahko na isti testni sliki, a na različnih mestih, razpoznamo dva objekta. V teh primerih so bile že učne slike take narave, da pri nekaterih zornih kotih ni bilo prekrivanja med objekti v učnih slikah (slika 6.13). Takim primerom bi se izognili tako, da bi objekte oz. njihova težišča na učnih slikah premaknili na sredino slike. Seveda se tak postopek ne bi obnesel za objekte nekaterih oblik, npr. objektov z obliko kolobarja.



Slika 6.13: Primer, ko preseka dveh hiperkrogel ne računamo. Presek maske pelikana in duhca je zelo majhen. V takem primeru lahko na isti sliki, a ne na istem mestu, najdemo dva različna objekta.

Gradnja hierarhije

Za vse pare iz kritične skupine izračunamo hierarhijo mask $H_0^{obj1 \& obj2}, \dots, H_6^{obj1 \& obj2}$. Maske apliciramo na ustrezne učne množice $U_0^{obj1 \& obj2}, \dots, U_6^{obj1 \& obj2}$, ki smo jih dobili pri

delitvi z upoštevanjem velikosti mask pri dvorazrednih SVM-jih.

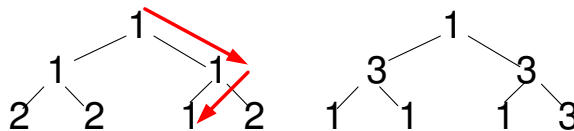
Za pare maskiranih učnih množic $U_0^{obj1} : U_0^{obj2}, \dots, U_0^{obj1} : U_6^{obj2}$ izračunam binarne SVM-je. Za vsak kritični par dobimo hierarhijo binarnih SVM-jev, ki jo uporabljamo v potrditveni fazi.

Potrditvena faza

Ko na sliki prepoznamo predmet s pomočjo hierarhije enorazrednih SVM-jev, je potrebno v primeru, da razpoznani objekti pripadajo kritičnim parom, razpoznavo še potrditi.

Primer potrditve: Na testni sliki razpoznamo *objekt 1* na podlagi hierarhije enorazrednih SVM-jev. Potrditev razpoznave mora biti veljavna za *objekt 1* v vseh hierarhijah binarnih SVM-jev, kjer se *objekt 1* pojavlja kot eden izmed objektov iz kritičnega para. Razpoznavo za *objekt 1* tudi v primeru binarnih SVM-jev potrdimo, ko obstaja konsistentna pot od korena pa do listov, ki glasuje za *objekt 1*. Slika 6.14 prikazuje primer potrditvene faze, ko *objekt 1* pripada dvema kritičnima paroma. Poudariti je potrebno, da moramo testno sliko pred preverjanjem tudi tu, tako kot v primeru enorazrednih SVM-jev, najprej ustrezno maskirati. To pomeni, da maskiramo vedno z isto masko kot v fazi učenja.

Celoten potek razpoznave je zapisan v obliki algoritma (algoritem 1).



Slika 6.14: Primer potrditvenega postopka za *objekt 1*. Recimo, da *objekt 1*, ki smo ga razpoznali v hierarhiji enorazrednih SVM-jev, tvori kritični par z *objektom 2* in *objektom 3*. Potrebno je izvesti potrditvena postopka *obj1:obj2* in *obj1:obj3*. V primeru *objekta 2* potrditveni postopek sicer potrdi razpoznavo *objekta 1*, vendar pa jo zavrne v primeru *objekta 3*. Potrditvena faza torej ne uspe in *objekt 1* na sliki ni bil razpoznan.

Algoritem 1 Faza razpoznavanja pri hierarhični SVM-metodi

/*Objekt na testni sliki najprej iščemo s pomočjo hierarhije enorazrednjih SVM-jev.*/
if Objekt, ki ga iščemo je razpoznan. **then**

if Objekt se nikoli ne pojavi v kritičnih parih. **then**

 Objekt je razpoznan.

else

 Potrebna je potrditev v hierarhijah kritičnih parov.

if Objekt je potrjen za vse hierarhije kritičnih parov, kjer se pojavlja. **then**

 Objekt je razpoznan.

else

 Zavržemo možnost, da je objekt, ki ga iščemo, na sliki.

end if

end if

else

 Objekta na sliki ni.

end if

Poglavje 7

Testi

7.1 Baza figuric

Bazo figuric sestavljajo štirje različni objekti: čajnik, duhec, miška in pelikan. Slike so bile zajete tako, da smo posamezen objekt postavili na rotacijsko mizico s črnim ozadjem in sliko zajemali na vsakih 5° . Razdalja med kamero in objektom je bila konstantna. Tako smo zajeli 72 RGB slik za vsak objekt. Za učno množico smo vzeli slike, zajete pri rotacijah 0° , 10° , 20° ,... V testno sliko smo vključili vse zajete slike, kar je smiselno, saj smo pri testiranju spreminjali ozadje.

Objekte na slikah smo segmentirali s pomočjo enostavnega upravljanja in ozadje obarvali popolnoma črno, vse tri RGB komponente so 0. Slike smo obrezali na isto velikost, in sicer tako, da se predmeti nahajajo na sredini slike, pri tem smo ohranili konstantno razdaljo med kamero in objekti. Nekaj slik iz baze figuric je na sliki 7.1. Pri poskusih smo uporabljali barvne RGB slike, velikosti 32×32 . Vhodni prostor ima tako 3072 značilk.



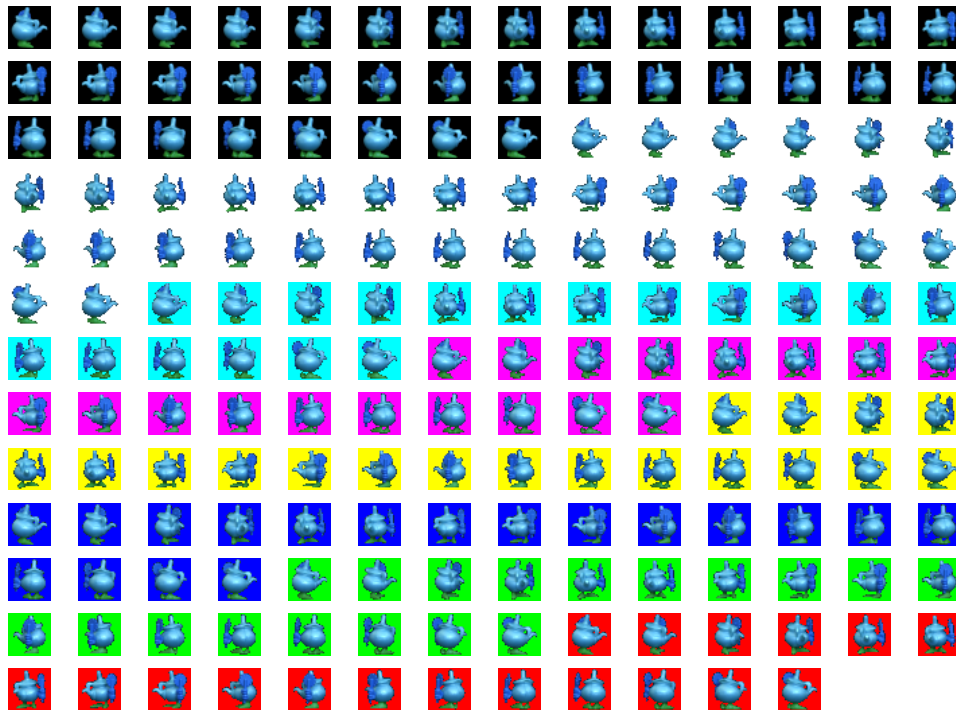
Slika 7.1: Nekaj slik iz baze figuric

7.2 Testiranje barvne BW-metode

7.2.1 Učna faza

Učno množico smo razširili po principu barvne BW-metode. Najprej smo za vsako učno sliko generirali slike z belim ozadjem in tako podvojili učno množico. Nato pa smo za vsako drugo učno sliko generirali še ozadja z ekstremnimi RGB vrednostmi: turkizne (0, 255, 255), škrlatne (255, 0, 255), rumene (255, 255, 0), modre (0, 0, 255), zelene (0, 255, 0) in rdeče (255, 0, 0) barve. Velikost učne množice smo 5-krat povečali, kar pomeni, da je velikost učne množice za posamezen objekt v konkretnem primeru 180 slik. Celotna učna množica za čajnik je na sliki 7.2. Generirali smo tudi množico slik, ki ne predstavljajo nobenega izmed objektov iz baze figuric. Tako smo poleg razredov čajnik, miška, duhec in pelikan dobili še razred, ki predstavlja ozadje oz. vse tiste slike, ki ne spadajo v nobenega izmed zgoraj naštetih razredov. Slike ozadja smo generirali iz naključno izbranih slik, ki niso vsebovale znanih objektov. Množica ozadja je vsebovala 477 slik. Nekaj izmed slik ozadja je na sliki 7.3.

V fazi učenja smo SVM-je zgradili za vse pare razredov, torej 10 SVM-jev. Vsi pari so bili linearno ločljivi, zato uporaba jedrnih funkcij ni bila potrebna.



Slika 7.2: Primer učne množice za barvno BW-metodo



Slika 7.3: Nekaj slik razreda ozadja

7.2.2 Testna faza in rezultati

Kot testne slike smo uporabili vseh 72 zajetih slik za posamezen objekt iz baze figuric. Učinkovitost metod smo preverjali tako, da smo generirali 11 različnih ozadij za vsako sliko, poleg belega in črnega ozadja še nekatere druge, ki so prikazani na sliki 7.4.

Za klasifikacijo v več razredov smo izbrali metodo teniškega turnirja. Različna ozadja smo označili z *bg1* do *bg9* ter *bgBlack* in *bgWhite*. Slike klasificiramo v 5 razredov z oznakami *r1* za čajnik, *r2* za duhca, *r3* za miško, *r4* za pelikana in *bg* za razred ozadja.

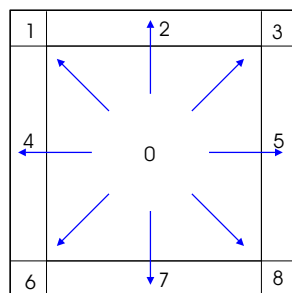


Slika 7.4: Primeri testnih slik na različnih ozadjih

Rezultati za vsako ozadje in razred posebej so prikazani v tabeli 7.1.

7.2.3 Občutljivost na premik

Vse teste iz prejšnjega razdelka smo ponovno izvedli z razširitvijo testne množice. Vsako testno sliko smo namreč premaknili za en slikovni element v 8 sosednjih lokacij: 1-zgoraj levo, 2-zgoraj, 3-zgoraj desno, 4-levo, 5-desno, 6-spodaj levo, 7-spodaj in 8-spodaj desno. Izhodiščno pozicijo slike označimo z 0. Shematski prikaz premika je na sliki 7.5. Da ohranimo sliko velikosti 32×32 , manjkajoči del premaknjene slike zapolnimo s črnimi slikovnimi elementi.



Slika 7.5: 8 smeri premikanja slike

ozadje	slike	r1	r2	r3	r4	bg	št. napak	delež razpoznav
bg1	čajnik	72	0	0	0	0	0	100,00
	duhec	0	72	0	0	0	0	100,00
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg2	čajnik	72	0	0	0	0	0	100,00
	duhec	0	65	0	0	7	7	90,28
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg3	čajnik	72	0	0	0	0	0	100,00
	duhec	0	42	0	0	30	30	58,33
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg4	čajnik	72	0	0	0	0	0	100,00
	duhec	0	62	0	0	10	10	86,11
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg5	čajnik	72	0	0	0	0	0	100,00
	duhec	0	72	0	0	0	0	100,00
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg6	čajnik	72	0	0	0	0	0	100,00
	duhec	0	72	0	0	0	0	100,00
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg7	čajnik	72	0	0	0	0	0	100,00
	duhec	0	72	0	0	0	0	100,00
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg8	čajnik	72	0	0	0	0	0	100,00
	duhec	0	59	0	0	13	13	81,94
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg9	čajnik	72	0	0	0	0	0	100,00
	duhec	0	72	0	0	0	0	100,00
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bgBlack	čajnik	72	0	0	0	0	0	100,00
	duhec	0	72	0	0	0	0	100,00
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bgWhite	čajnik	72	0	0	0	0	0	100,00
	duhec	0	71	0	0	1	1	98,61
	miška	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00

Tabela 7.1: Rezultati za barvno BW-metodo pri raznolikem ozadju. V prvem stolpcu je vpisana oznaka ozadja, v drugem je ime predmeta, ki je na testni sliki. Sledijo stolpci z oznakami razredov, kamor je klasifikator klasificiral slike. V predzadnjem je število napačno klasificiranih slik za posamezen predmet. V zadnjem stolpcu je delež pravilno razpoznanih podan v odstotkih. Vrstico duhec za ozadje bg2 si torej razložimo na naslednji način: 65 slik duhca, ki se nahaja na ozadju z oznako bg2, je bilo uvrščenih v razred r2, torej pravilno, preostalih 7 slik pa v razred ozadja, kar je narobe. Število napak je v tem primeru 7. Delež razpoznav za slike čajnika na ozadju bg2 pa je 90,28%.

Rezultati testiranja na premik v primeru barvne BW-metode so v tabelah 7.2, 7.3 in 7.4. Opazimo lahko, da je metoda manj občutljiva na premik levo-desno kot na premik navzgor-navzdol. To je nekako logično in je posledica rotacije objekta okrog z osi (glede na rotacijsko mizico), saj smo tudi v učni fazi dopuščali več sprememb levo-desno, kot pa navzgor-navzdol.

BW- metoda	št. pravilno klasificiranih/št. napačno klasificiranih (glede na pozicijo premaknjene slike)		
	2496/672	2703/465	2453/715
	2994/174	3107/ 61	2869/299
	2640/528	2911/257	2392/776

Tabela 7.2: Tabela prikazuje občutljivost barvne BW-metode na premik. Pozicija v tabeli pomeni smer premika. Tako rezultat v zgornjem levem kotu pomeni test na slikah, ki smo jih premaknili za en slikovni element levo in za en slikovni element navzgor. Na vsakem področju smo imeli 3168 slik = $72(\text{slik za posamezen objekt}) \times 4(\text{objekti}) \times 11(10 \text{ generiranih ozadij} + \text{črno})$.

BW- metoda	delež pravilno razpoznanih slik v % glede na pozicijo premaknjene slike		
	78,79%	85,32%	77,43%
	94,51%	98,07%	90,56%
	83,33%	91,89%	75,51%

Tabela 7.3: Tabela prikazuje delež razpoznavne barvne BW-metode glede na premik. Pozicija pomeni smer premika (podobno kot v tabeli 7.2). Delež razpoznavne v % = število pravilno razpoznanih/število vseh testnih slik.

BW- metoda	TP	FP
	86,16%	13,84%

Tabela 7.4: Rezultati za BW-metodo — skupno za vsa ozadja in vse premike

7.3 Testiranje hierarhičnega pristopa

7.3.1 Učna faza

Razširitev testnih slik z zamikanjem

Tudi pri hierarhičnem pristopu smo povečali učno množico. Nove učne (in testne) slike smo generirali tako, da smo objekt na sliki premikali v vse možne smeri za en slikovni element (slika 7.5). Tako smo generirali za vsako posamezno sliko še 8 dodatnih. Velikost tako generiranih učnih množic je 324 slik za posamezen razred. Ker smo premike vključili že v učenje, smo v primeru hierarhičnega SVM-ja vedno testirali na vseh slikah, premaknjenih in nepremaknjenih.

Za vsak objekt (čajnik, duhec, miška in pelikan) smo izračunali hierarhijo enorazrednih SVM-jev in za kritične pare tudi hierarhijo dvorazrednih SVM-jev.

7.3.2 Testna faza in rezultati

Na vsaki testni sliki smo iskali vsak učni objekt iz baze figuric posebej. Torej štiri preverjanja za vsako testno sliko.

Za vsak objekt na izbranem ozadju smo imeli 648 slik ($72 \text{ slik} \times 9 \text{ pozicij}$). Tabela 7.5 vsebuje rezultate za vsako posamezno ozadje posebej. Rezultati za hierarhični SVM po posameznih področjih, glede na delež pravilno razpoznanih in glede na vse testne slike so zbrani v tabelah od 7.6 do 7.8.

Medtem ko je bila pri BW-pristopu vsaka slika razporejena v natanko en razred, v primeru hierarhičnega pristopa to ni nujno. Lahko se zgodi, da slika ni razporejena v nobenega izmed 4 razredov ali pa v več kot en razred. Posledica tega je, da vsota TP + FP ni enaka številu testiranih slik.

hierarhični SVM	r1		r2		r3		r4		vsi 4 razredi skupaj	
	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
bg1	607	0	647	0	622	0	550	13	2426/2592(93,60%)	13(0,50%)
bg2	644	0	648	0	642	0	565	86	2499/2592(96,41%)	86(3,32%)
bg3	630	0	648	0	647	0	488	8	2413/2592(93,09%)	8(0,31%)
bg4	512	0	600	0	647	0	375	0	2134/2592(82,33%)	0(0,00%)
bg5	587	0	641	0	538	0	558	0	2324/2592(89,66%)	0(0,00%)
bg6	509	0	581	0	522	0	560	8	2172/2592(83,80%)	8(0,31%)
bg7	562	0	648	0	642	0	227	0	2079/2592(80,21%)	0(0,00%)
bg8	551	0	645	0	647	0	334	1	2177/2592(83,99%)	1(0,04%)
bg9	598	0	638	0	604	0	362	6	2202/2592(84,95%)	6(0,23%)
bgWhite	602	0	635	0	647	0	319	0	2203/2592(84,99%)	0(0,00%)
bgWlack	644	0	645	0	638	0	615	10	2542/2592(98,07%)	10(0,39%)
vse skupaj	6446	0	6976	0	6796	0	4953	132	25171(88,28%)	132(0,46%)

Tabela 7.5: Rezultati za hierarhično metodo pri raznolikem ozadju. Za vsak objekt na izbranem ozadju smo imeli 648 ($72 \text{ slik} \times 9 \text{ pozicij}$) slik, kar pomeni 2592 testnih slik za vsako posamezno ozadje. Tabela kaže, koliko objektov je bilo pravilno klasificiranih (TP - true positives) in koliko jih je bilo napačno klasificiranih (FP - false positives).

Delež razpoznavne je pri hierarhičnem pristopu v nekaterih primerih manjši od deleža razpoznavne pri BW-metodi, kar je v veliki meri posledica pozicioniranja testne maske. Če bi testno masko porinili še nekoliko čez rob slike, bi gotovo v nekaterih primerih dobili večjo stopnjo razpoznavne. Poleg tega je hierarhična metoda precej občutljiva na šum in prekrivanja. Posledica tega je, da napačno klasifikacijo v nekaterih primerih povzroči že nekoliko površno zamenjano ozadje, to je, če poleg ozadja spreminjamo tudi slikovne elemente, ki sicer pripadajo objektu. V nekaterih primerih, ko tudi objekti vsebujejo slikovne elemente črne barve, namreč tudi te zamenjamo skupaj s slikovnimi elementi ozadja, kar pa lahko vodi k napačni klasifikaciji.

hierarhični SVM	št. pravilno klasificiranih/št. napačno klasificiranih glede na pozicijo premaknjene slike		
	2645/26	2758/24	2531/23
	2975/15	2973/11	2921/10
	2763/10	2886/6	2719/7

Tabela 7.6: Rezultati za hierarhično metodo pri raznolikem ozadju po področjih. Področje je nakazano s pozicijo v tabeli: npr. zgoraj levo so podatki o slikah, ki so bile premaknjene za en slikovni element navzgor in levo. Na vsakem področju smo imeli 3168 slik = 72(slik za posamezen objekt) \times 4(objekti) \times 11(10 generiranih ozadji+črno).

hierarhični SVM	delež pravilno razpoznanih slik v% glede na pozicijo premaknjene slike		
	83,49%	87,06%	79,89%
	93,91%	93,84%	92,20%
	87,22%	91,10%	85,83%

Tabela 7.7: Tabela prikazuje delež razpoznave hierarhičnega SVM-ja glede na premik. Pozicija pomeni smer premika (podobno kot v tabeli 7.2). Delež razpoznave v % = število pravilno razpoznanih/število vseh objektov.

hierarhični SVM	TP	FP
	88,28%	0,46%

Tabela 7.8: Rezultati za hierarhično metodo — vse skupaj

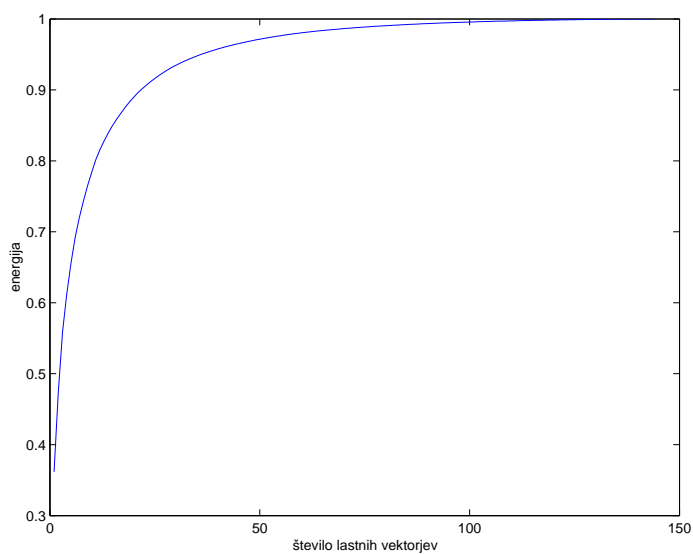
7.4 Testiranje robustnega PCA pristopa

7.4.1 Učna faza

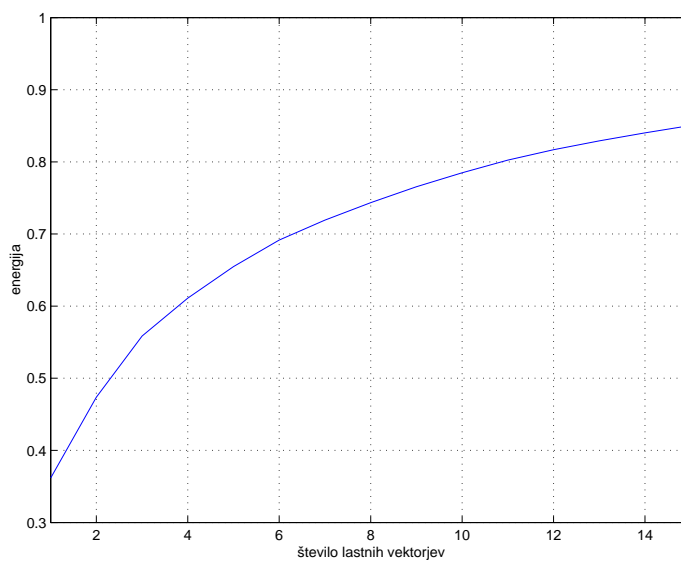
Iz vseh učnih slik objektov (36 slik za vsak objekt) zgradimo lastni prostor. Izračunamo lastne vrednosti in lastne vektorje. Pri odločitvi o številu lastnih vektorjev p , s katerimi bomo predstavili podatke, si pomagamo z deležem energije lastnih vektorjev (slika 7.6). Delež energije prvih p lastnih vektorjev izrazimo kot vsoto prvih p lastnih vrednosti deljeno z vsoto lastnih vrednosti vseh lastnih vektorjev. Običajno je potrebno za zadovoljivo predstavitev potrebno nekaj več kot 70% energije, kar v našem poskusu pomeni okrog 7 lastnih vektorjev. Na podlagi krivulje (slika 7.6), ki predstavlja energijo lastnih vrednosti lastnega prostora figuric, in eksperimentov, smo se odločili, da bomo za predstavitev uporabili 8 lastnih vektorjev (slika 7.7), kar pomeni 74% energije. Manjše število lastnih vektorjev bi vodilo v presiromašno predstavitev objektov. Če vzamemo premalo vektorjev, slik ne predstavimo dovolj dobro. Prav tako ni dobro, da vzamemo preveč lastnih vrednosti, saj se lahko zgodi, da modeliramo tudi šum, kar pa posledično lahko privede do napak pri klasifikaciji.

7.4.2 Testna faza in rezultati

Metodo najprej testiramo na slikah z različnim ozadjem. Test nato razširimo tudi na premaknjene slike. Za vsako sliko generiramo 18 hipotez. Slikovnih elementov ne izbiramo čisto naključno. Izbiranje podmnožic usmerimo z maskami, ki smo jih uporabljali že v hierarhičnem pristopu. Vsako hipotezo zgradimo na osnovi ene izmed mask. Maske so prikazane na sliki 7.8 in dejansko predstavljajo: presek in unijo učnih slik pelikana ter celotno hierarhijo učnih slik duhca. Lahko bi uporabili tudi druge kombinacije mask, vendar so eksperimenti pokazali, da je bil tudi opisan izbor dovolj dober. Za vsako izbrano masko generiramo po dve hipotezi. Uporabljene maske ustrezno prekrijejo vsa področja tako, da lahko uspešno generiramo pravilne hipoteze. Pri robustnem generiranju hipotez vedno vzamemo vse tri RGB komponente za posamezni slikovni element. Ni možno, da bi generirali hipotezo npr. le na podlagi R in G komponente nekega slikovnega elementa. Podmnožico slikovnih elementov vedno izberemo iz belega dela maske.

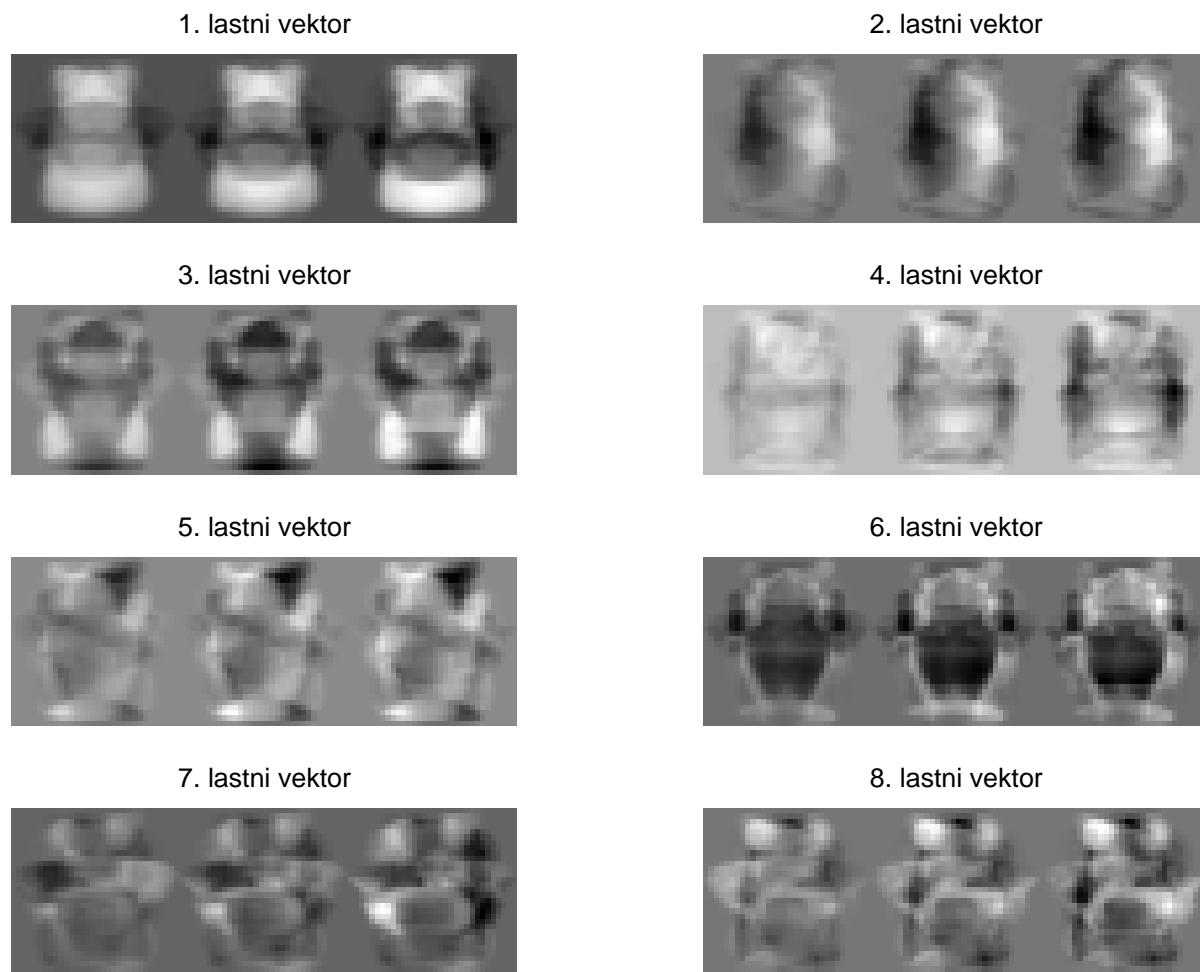


(a)

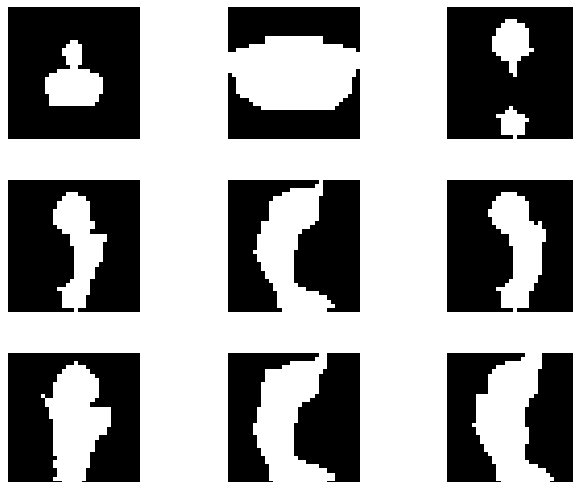


(b)

Slika 7.6: Energija lastnih vektorjev: (a) energija za vse možne velikosti podprostora, (b) energija za velikosti podprostorov $p \leq 15$ na povečani skali



Slika 7.7: Prvih 8 lastnih vektorjev glede na velikost lastnih vrednosti. Vsaka barvna komponenta je prikazana posebej. Prikaz je za vsako sliko posebej prilagojen tako, da minimalna vrednost predstavlja črne pike, maksimalna pa bele. Tak način omogoča, da so slike dobro vidne. Vendar pa to pomeni, da isti odtenek na različnih slikah ne predstavlja iste vrednosti lastne slike.



Slika 7.8: Področja izbire slikovnih elementov

V prvi iteraciji hipotezo zgradimo na podlagi 80 slikovnih elementov oz. 240 značilkah RGB vektorja slike. V iterativnem postopku zmanjšujemo začetno množico slikovnih elementov s faktorjem $\alpha = 0,9$ tako, da odstranimo del slikovnih elementov z največjo napako. Zadnja iteracija vsebuje vsaj 56 točk. Parametre, ki smo jih uporabili pri robustnem PCA-ju, smo določili eksperimentalno.

Sliki 7.9 in 7.10 predstavljata dve testni sliki njune hipoteze. Testni sliki v skrajnem zgornjem levem kotu sledi rekonstrukcija ostalih 18 generiranih hipotez. Vse generirane hipoteze niso dobre in od vseh obdržimo le najboljšo, ki jo uporabimo tudi za klasifikacijo slike. Za dani problem se je kot najboljši kriterij za izbiro hipoteze izkazalo povprečje kvadratov napak rekonstrukcij ξ_i na točkah, iz katerih smo generirali hipotezo.

Za izbiro najboljše hipoteze smo uporabili naslednjo mero:

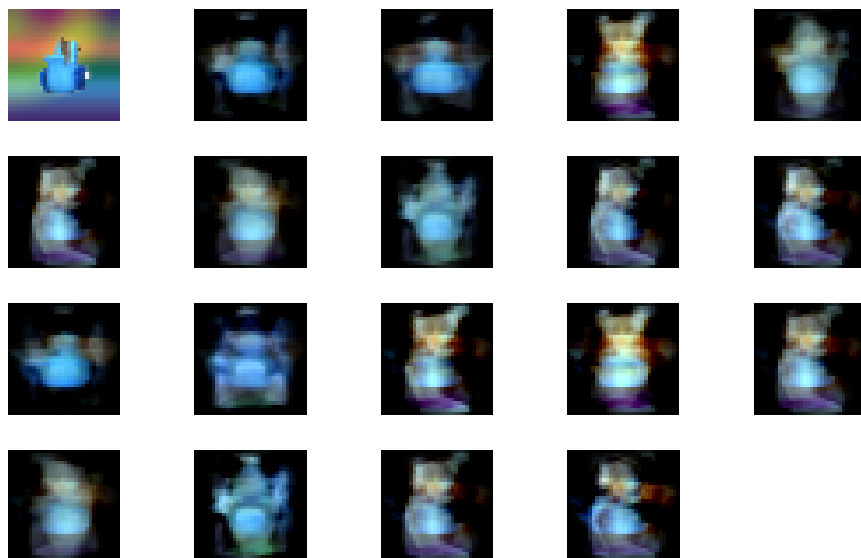
$$q(i) = \frac{\sum_{\mathcal{K}_i} \xi_j^2}{|\mathcal{K}_i|}, \quad (7.1)$$

kjer označuje $|\mathcal{K}_i|$ število izbranih značilk v RGB sliki in ξ_j rekonstrukcijsko napako značilke. Izbrana je hipoteza z najmanjšo vrednostjo $q(i)$.

Za vsako hipotezo i , $i = 1, \dots, 18$, izračunamo vrednost kriterijske funkcije $q(i)$. Izberemo tisto hipotezo \mathbf{a}_i , ki ima najmanjšo vrednost funkcije q , torej najmanjšo povprečno napako nad točkami, ki smo jih uporabili pri računanju \mathbf{a}_i . Tako v primeru iz slike 7.9



Slika 7.9: Zgornja leva slika prikazuje testno sliko. Ostalih 18 predstavlja vse generirane hipoteze za dano sliko.



Slika 7.10: Zgornja leva slika prikazuje testno sliko. Ostalih 18 predstavlja vse generirane hipoteze za dano sliko.

izberemo prvo generirano hipotezo. Njeno rekonstrukcijo prikazuje druga slika v prvi vrstici. V drugem primeru pa izberemo 16. hipotezo. Njena rekonstrukcija je druga slika v zadnji vrstici (slika 7.10).

Glede na izbrano hipotezo vsako posamezno testno sliko klasificiramo. Uporabili smo dva različna klasifikatorja: najbližji sosed — NN in metodo podpornih vektorjev — SVM (tabela 7.9).

robustni PCA	ozadje	NN-klasifikator				SVM-klasifikator			
		r1	r2	r3	r4	r1	r2	r3	r4
	bg1	0	0	0	0	0	0	0	0
	bg2	0	0	0	2	0	0	0	2
	bg3	0	0	0	0	0	0	0	2
	bg4	0	0	0	0	0	0	0	0
	bg5	0	0	0	0	0	0	0	0
	bg6	0	0	0	0	0	0	0	0
	bg7	0	0	0	0	0	0	0	0
	bg8	0	0	0	1	0	0	0	2
	bg9	0	0	0	0	0	0	0	0
	bgBlack	0	0	0	0	0	0	0	0
	bgWhite	0	0	0	0	0	0	0	1

Tabela 7.9: Primerjava NN- in SVM-klasifikatorja za robustno PCA-metodo. Tabela prikazuje število napak za vsak klasifikator, vsak razred in vsako ozadje posebej. Testi so narejeni na nezamaknjenih slikah.

Pri klasifikatorju NN testno sliko klasificiramo v razred, ki mu pripada učna slika z vektorjem koeficientov \mathbf{a} , ki je najbližja koeficientom testne slike. Pri metodi SVM zgradimo dvorazredne SVM-je na podlagi koeficientov \mathbf{a} učne množice. Rezultati obeh klasifikatorjev so prikazani v tabeli 7.9. Opazimo, da ni bistvene razlike med obema klasifikatorjema. Še celo nekoliko nižji odstotek razpoznavne ima SVM-klasifikator. V danem primeru je torej vseeno, kateri klasifikator uporabimo. Iz projekcij v 2-D in 3-D lastni podprostor (sliki 7.11 in 7.12) je jasno, da bo NN-klasifikator deloval dobro, saj so množice posameznih objektov med seboj zelo dobro ločene. Za konkreten primer je torej NN-klasifikator boljši ali pa enakovreden SVM-klasifikatorju. SVM-klasifikator bi

se izkazal za boljšega, če točke projekcij, ki pripadajo posameznim objektom, ne bi bile dobro ločljive med seboj. V takem primeru bi bilo verjetno najbolje, če bi se poslužili nelinearnega SVM-ja.

Zaradi enakovrednosti obeh klasifikatorjev, v tabeli 7.10 podajamo podrobno razčlenitev rezultatov le za primer NN-klasifikatorja.

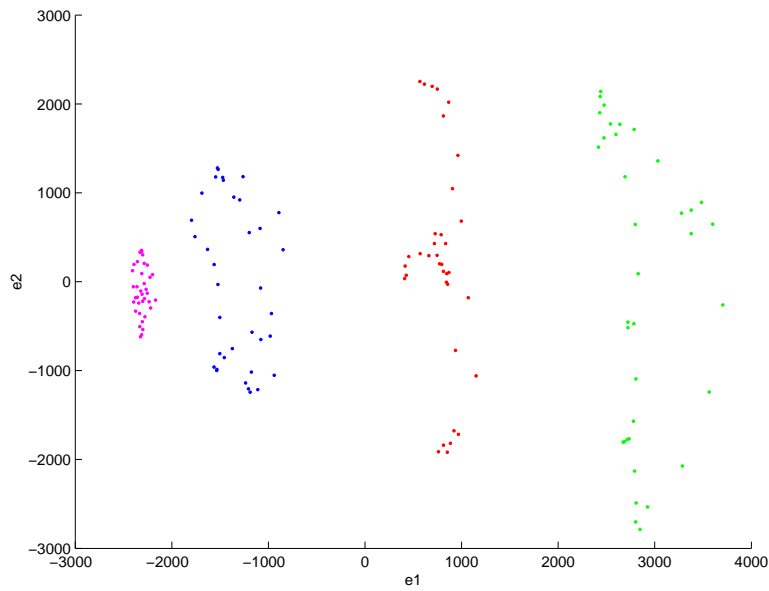
SVM je boljši s stališča hitrosti in prostorske kompleksnosti. V primeru linearnega SVM-ja je za klasifikacijo potreben le enostaven izračun ene same funkcije (2.1). V primeru uporabe nelinearnih problemov je izračun sicer nekoliko kompleksnejši in je časovno odvisen od števila podpornih vektorjev. Vendar pa obstajajo rešitve [2], ki množico podpornih vektorjev nadomestijo z manjšo množico umetno generiranih vektorjev. Tako zmanjšajo časovno in tudi prostorsko kompleksnost SVM-klasifikatorja, saj odločitveno funkcijo SVM določa le množica tistih vhodnih podatkov, ki predstavljajo podporne vektorje. Pri NN za klasifikacijo vedno potrebujemo celotno učno množico. Učne primere sicer lahko uredimo v učinkovite strukture npr. r-drevesa [7], ki omogočajo hitro iskanje najbližjega primera, vendar pa kompleksnost raste s številom učnih primerov, kar pa pri SVM-jih ni nujno.

7.4.3 Občutljivost na premik

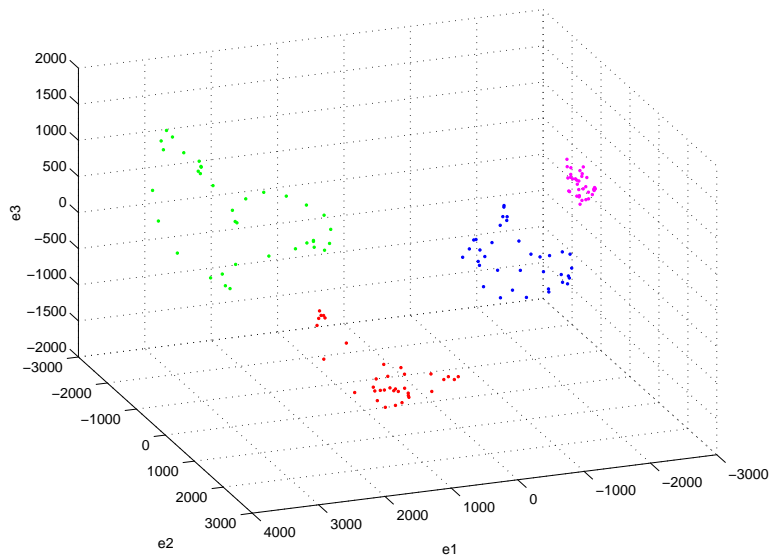
Na enak način kot v prejšnjih testih tudi tu premaknemo testne slike v vseh 8 možnih smereh. Za učno množico vzamemo vseh 36 slik posameznega predmeta. Tudi tu primerjamo NN- in SVM-klasifikator (tabele 7.11, 7.12 in 7.13). Tudi v primeru premika se klasifikatorja izkažeta za enakovredna.

ozadje	slike	r1	r2	r3	r4	bg	št. napak	delež razpoznavne
bg1	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg2	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	0	2	0	70	0	2	97,22
bg3	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	2	0	1	69	0	3	95,83
bg4	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg5	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg6	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg7	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg8	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bg9	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bgBlack	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	0	0	0	72	0	0	100,00
bgWhite	cajnik	72	0	0	0	0	0	100,00
	strahec	0	72	0	0	0	0	100,00
	miska	0	0	72	0	0	0	100,00
	pelikan	2	0	0	70	0	2	97,22

Tabela 7.10: Rezultati za robustno PCA-metodo pri raznolikem ozadju. Podobno kot v tabeli 7.1 je v prvem stolpcu vpisana oznaka ozadja, v drugem je ime predmeta, ki je na testni sliki. Sledijo stolpci z oznakami razredov, kamor smo klasificirali slike. V predzadnjem je število napačno klasificiranih slik in v zadnjem stolpcu delež pravilno razpoznanih slik v %.



Slika 7.11: Projekcije slik objektov v podprostor, določen s prvima dvema lastnima vektorjema. Modre točke pripadajo čajniku, rdeče duhcu, zelene miški in roza pelikanu.



Slika 7.12: Projekcije slik objektov v podprostor, določen s prvimi tremi lastnimi vektorji. Modre točke pripadajo čajniku, rdeče duhcu, zelene miški in roza pelikanu.

robustni PCA	Rezultati za robustno PCA-metodo po področjih					
	NN-klasifikator			SVM-klasifikator		
	(TP/FP)			(TP/FP)		
	3069/ 99	3091/ 77	3058/110	3058/110	3069/ 99	3036/132
	3157/ 11	3168/ 0	3157/ 11	3135/ 33	3146/ 22	3146/ 22
	2937/231	3025/143	2981/187	2937/231	3025/143	2981/187

Tabela 7.11: Rezultati za hierarhično metodo pri raznolikem ozadju po področjih. Področje je nakazano s pozicijo v tabeli: npr. zgoraj levo so podatki o slikah, ki so bile premaknjene za en slikovni element navzgor in levo. Na vsakem področju smo imeli 3168 slik = $72(\text{slik za posamezen objekt}) \times 4(\text{objekti}) \times 11(10 \text{ generiranih ozadij} + \text{črno})$. Rezultati so podani za oba klasifikatorja: NN (levo) in SVM (desno).

robustni PCA	delež pravilno razpoznanih slik v % za robustno PCA-metodo po področjih					
	NN-klasifikator			SVM-klasifikator		
	(TP/FP)			(TP/FP)		
	96,88%	97,57%	96,53%	96,53%	96,88%	95,83%
	99,65%	100,00%	99,65%	98,96%	99,31%	99,31%
	92,71%	95,49%	94,10%	92,71%	95,49%	94,10%

Tabela 7.12: Rezultati za robustno PCA-metodo pri raznolikem ozadju po področjih. Področje je nakazano s pozicijo v tabeli: npr. zgoraj levo so podatki o slikah, ki so bile premaknjene za en slikovni element navzgor in levo. Zgornja tabela(TP/FP). Na vsakem področju smo imeli $3168 = 72(\text{slik}) \times 4(\text{objekti}) \times 11(10 \text{ generiranih ozadij} + \text{črno})$. Rezultati so podani za oba klasifikatorja: NN (levo) in SVM (desno).

robustni PCA	NN-klasifikator		SVM-klasifikator	
	TP	FP	TP	FP
	96,95%	3,05%	96,57%	3,43%

Tabela 7.13: Povzetek rezultatov za robustno PCA-metodo, za vsak klasifikator posebej

7.5 Povzetek testiranj metod

Še enkrat strnimo rezultate vseh treh metod v skupni tabeli: 7.14 in 7.15.

	TP	FP
BW-metoda	86,16%	13,84%
hierarhični SVM	88,28%	0,46%
robustni PCA (NN)	96,95%	3,05%

Tabela 7.14: Rezultati za vse tri metode v celoti

	delež pravilno razpoznanih slik v % glede na pozicijo premaknjene slike		
BW- metoda	78,79%	85,32%	77,43%
	94,51%	98,07%	90,56%
	83,33%	91,89%	75,51%
hierarhični SVM	83,49%	87,06%	79,89%
	93,91%	93,84%	92,20%
	87,22%	91,10%	85,83%
robustni PCA (NN)	96,88%	97,57%	96,53%
	99,65%	100,00%	99,65%
	92,71%	95,49%	94,10%

Tabela 7.15: Primerjava deležev razpoznavne za vse tri metode. Pozicija pomeni smer premika (podobno kot v tabeli 7.2). Delež razpoznavne v % = število pravilno razpoznanih/število vseh objektov.

Če bi pogledali samo končne rezultate in primerjali metode le na podlagi števil, ne bi dobili objektivne ocene. Pomembno je poudariti in izpostaviti lastnosti posameznih metod. Učne množice so bile prirejene tako, da bi kar najbolj povečale sposobnost posameznega klasifikatorja. Šibkost BW-metode je občutljivost na ozadje, zato smo učno

množico barvnega BW-ja povečevali z različnimi ozadji. Hierarhičen SVM je občutljiv na premik, zato smo prvotno učno množico povečali s slikami, ki vključujejo tudi premike objektov na sliki.

Številke najnižje rangirajo BW-metodo, saj ima le-ta najnižjo stopnjo TP in najvišji odstotek FP. Vendar je tu predvsem pomembno poudariti, da je izmed vseh treh testiranih metod ravno ta najbolj enostavna. Najbolj težaven del je definicija razreda ozadja, vendar ko je ta enkrat določen, metoda od vseh treh deluje najhitreje. Če je okolica, v kateri se predmet pojavlja, poznana, lahko razred ozadja dobro definiramo in v tem primeru metoda zanesljivo deluje. Poleg tega imamo vedno možnost odločitvene funkcije ponovno izračunati na podlagi “bootstrappinga” in tako izboljšati sposobnost pravilne klasifikacije. V sistemih, kjer je hitrost bolj pomembna kot natančnost in kjer nas visoka stopnja FP ne moti, je mogoče BW-metoda najboljša izbira.

Kjer pa je visoka stopnja natančnosti bistvenega pomena, sta veliko boljše izbira hierarhični SVM in robustni PCA. Odstotek TP za hierarhični SVM v tabeli je sicer precej nižji kot za robustni PCA. Tu je potrebno povedati, da bi ta odstotek zvišali, če bi masko na testne slike ne samo položili, ampak poskušali najti tudi njen idealen položaj na sliki. Pravilna razpoznavna pri hierarhičnem SVM-ju je namreč precej odvisna od pravilne namestitve maske. Idealno je, če jo položimo tako, da zajema samo slikovne elemente, ki pripadajo objektu. Ker je hierarhični SVM občutljiv na šum znotraj maske, gre nekaj napačno klasifikacij pripisati tudi načinu spreminjanja ozadja. Ozadje namreč zamenjamo tako, da spremenimo vse slikovne elemente na sliki, ki imajo vrednost manj od nekega vnaprej določenega praga. Tako se lahko zgodi, da pošumljamo tudi slikovne elemente, ki pripadajo objektom. Metoda je sicer neobčutljiva na raznoliko ozadje, je pa občutljiva na spremembe znotraj objekta samega.

Lepa lastnost hierarhične SVM-metode je zelo nizka stopnja FP. To je posledica preverjanj iste slike s celo hierarhijo klasifikatorjev. Metodi bi lahko očitali nerodnost, saj je treba vsak objekt iskati z drugo hierarhijo klasifikatorjev, kar pomeni tudi veliko časovno zahtevnost. Je pa praktično to, da nam ni potrebno vedno preverjati cele hierarhije. Za večino slik, ki ne vsebujejo slik znanih elementov, zadostuje že preverjanje na prvem nivoju posamezne hierarhije, saj le-ta z veliko natančnostjo zavrže slike, ki ne vsebujejo iskanih objektov.

Časovna zahtevnost se pri hierarhičnem SVM-ju v najslabšem primeru linearno povečuje s številom objektov v bazi, medtem ko časovna zahtevnost BW-metode s številom objektov narašča logaritemsko. Časovna kompleksnost robustnega PCA-ja pa narašča s številom slik v učni množici. Kakšno je to naraščanje je odvisno od implementacije — v najslabšem primeru je linearno.

Robustni PCA ima podobne lastnosti kot hierarhičen PCA. Izbiro podmnožic slikovnih elementov v konkretnem primeru ravno tako usmerjamo z masko. Vendar tu ni potrebno, da je maska točno pozicionirana. Prav tako nam ni potrebno določiti mask za vsak objekt posebej, ampak le neko skupino mask, ki se izkažejo kot dobre za vse objekte. Poleg tega robustni PCA tolerira šum tudi znotraj maske, saj točke z veliko rekonstrukcijsko napako izločamo. V tem pogledu je boljši kot hierarhični SVM. V nekaterih primerih se preveč prilagodi le podmnožici točk, kar pomeni, da lahko najde objekt tudi tam, kjer ga ni. To se kaže v višji stopnji FP v primerjavi s hierarhičnim SVM-jem.

Tabela 7.15 razčleni rezultate metod glede na premik. Za vse tri metode velja, da so manj občutljive na premike levo-desno kot na premike navzgor-navzdol. Sklepamo, da je to posledica karakteristike množice učnih slik, kjer večjo variabilnost dopuščamo v vodoravni smeri, saj smo predmete na slikah zajeli tako, da smo jih rotirali okrog vertikalne osi.

Razlika med BW-metodo in hierarhičnim SVM-jem je tudi v načinu, kako posamezna metoda operira s prostorom značilk. Medtem ko pri BW poskušamo prostor čim bolj razmejiti, ga pri hierarhičnem SVM-ju najprej poskušamo čim bolj omejiti. V primeru, da se prostori posameznih predmetov med seboj prekrivajo, prostor poskušamo še razmejiti.

Na tem mestu je smiselno opozoriti tudi na specifične lastnosti objekta pelikan. Za razliko od ostalih predmetov se področje, ki ga zajema pelikan na sliki s spreminjanjem zornega kota, zelo spreminja. Pelikan od strani zavzema precej večji del slikovnih elementov na sliki kot pelikan, prikazan od spredaj ali od zadaj (slika 7.1). Prav zaradi majhega odstotka slikovnih elementov na nekaterih slikah pelikana je pelikan najtrši oreh za vse tri metode. Poleg tega je pelikah iz nekaterih pogledov zelo podoben osrednjemu delu čajnika, kar še dodatno oteži pravilno klasifikacijo.

7.6 Primerjava BW-, hierarhične SVM- in robustne PCA-metode na slikah večjih dimenzij

S testi v razdelkih 7.2, 7.3 in 7.4 smo testirali sposobnost ločevanja med objekti posameznih metod. V resničnem okolju pa se pojavi ne le potreba po ločevanju med objekti, temveč tudi potreba po lokalizaciji objekta na določenem mestu na sliki. V ta namen smo vse tri metode testirali še na treh večjih slikah. Zastavljena naloga zahteva tako klasifikacijo kot tudi lokalizacijo objektov na slikah. Sliki 7.13(a) in 7.13(b) vključujeta različne objekte iz različnih zornih kotov. Kompoziciji objektov na obeh slikah sta enaki, sliki se razlikujeta samo po tipu ozadja. V prvem primeru je ozadje črno, v drugem pa rdečkasto. Slika 7.13(c) prikazuje objekte, postavljene na realno ozadje. Testiranje je potekalo tako, da smo se z oknom v velikosti učnih slik (32×32) premikali po vseh možnih lokacijah na testnih slikah. Na vsaki lokaciji smo naredili test za vse tri metode.



(a) Objekti na črnem ozadju (b) Objekti na rdečkastem ozadju (c) Objekti na realnem ozadju

Slika 7.13: Testne slike

V primeru metode BW in hierarhičnega SVM-ja smo postopali enako kot v razdelkih 7.2 in 7.3. V primeru robustnega PCA-ja pa smo hipoteze za posamezno sliko ocenili na podlagi glasovalne funkcije (5.12). Vsak objekt posebej je zahteval določitev praga glasovalne funkcije. Za vsako posamezno lokacijo okna na sliki smo izbrali hipotezo z največ zbranimi glasovi in razpoznali objekt, če je glasovalna funkcija presegla postavljen prag. Pragove smo določili ročno na osnovi predhodnega eksperimentiranja. σ v enačbi

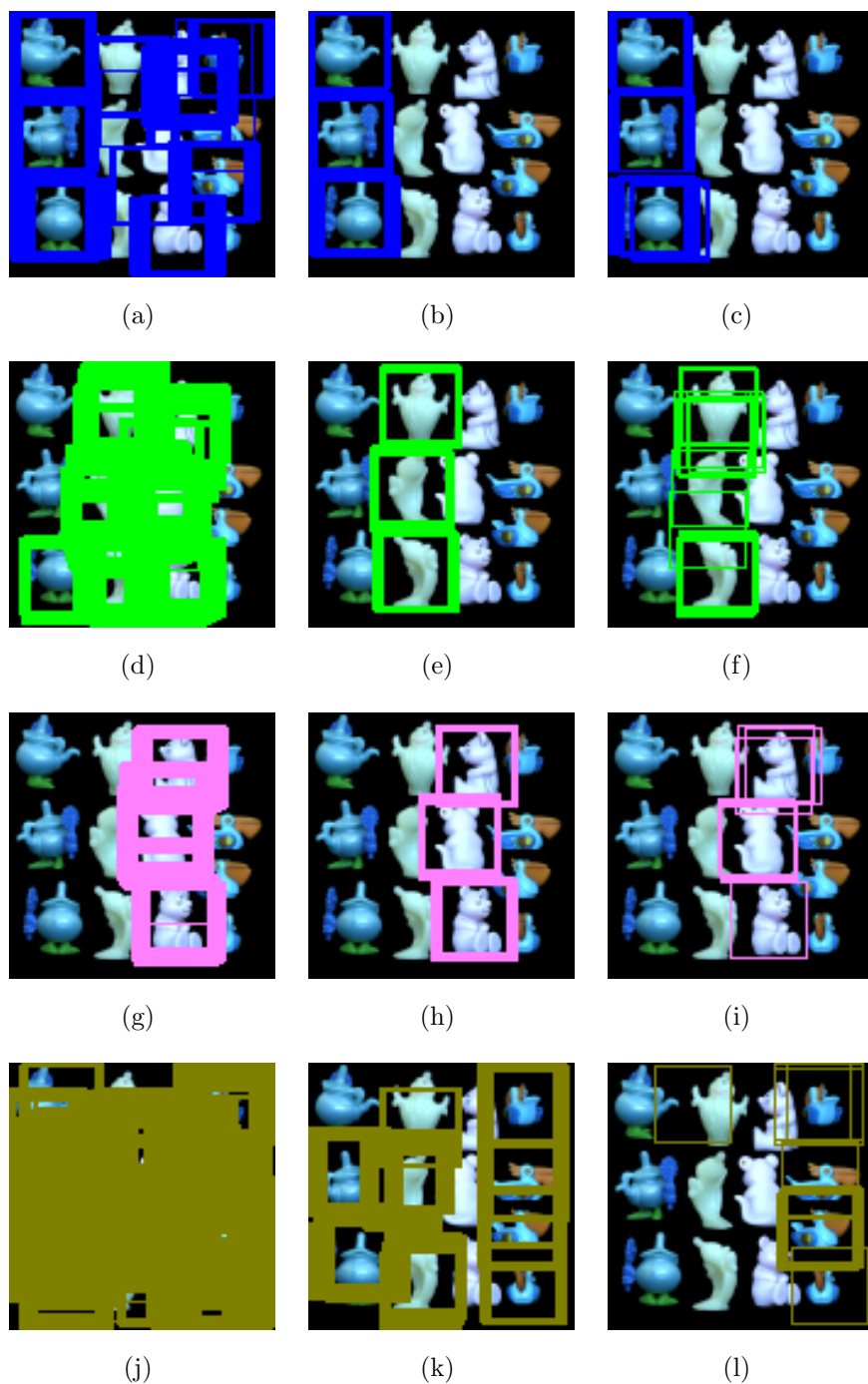
5.12 je zavzemala vrednost 0,2. Seveda so rezultati odvisni od izbire praga, vendar pa je očitno, da imajo tiste testne slike, ki v resnici vsebujejo objekte, za katere glasujejo, v splošnem precej večje vrednosti glasovalnih funkcij kot slike, ki prikazujejo ozadje. Če prag nižamo, nižamo tudi verjetnost, da nekega objekta na sliki ne bomo našli, vendar pa hkrati višamo FP. Prag smo nastavili tako, da smo uravnotežili razmerje med številom nerazpoznanih objektov in številom FP.

Rezultati so prikazani na slikah: 7.15, 7.16 in 7.17.

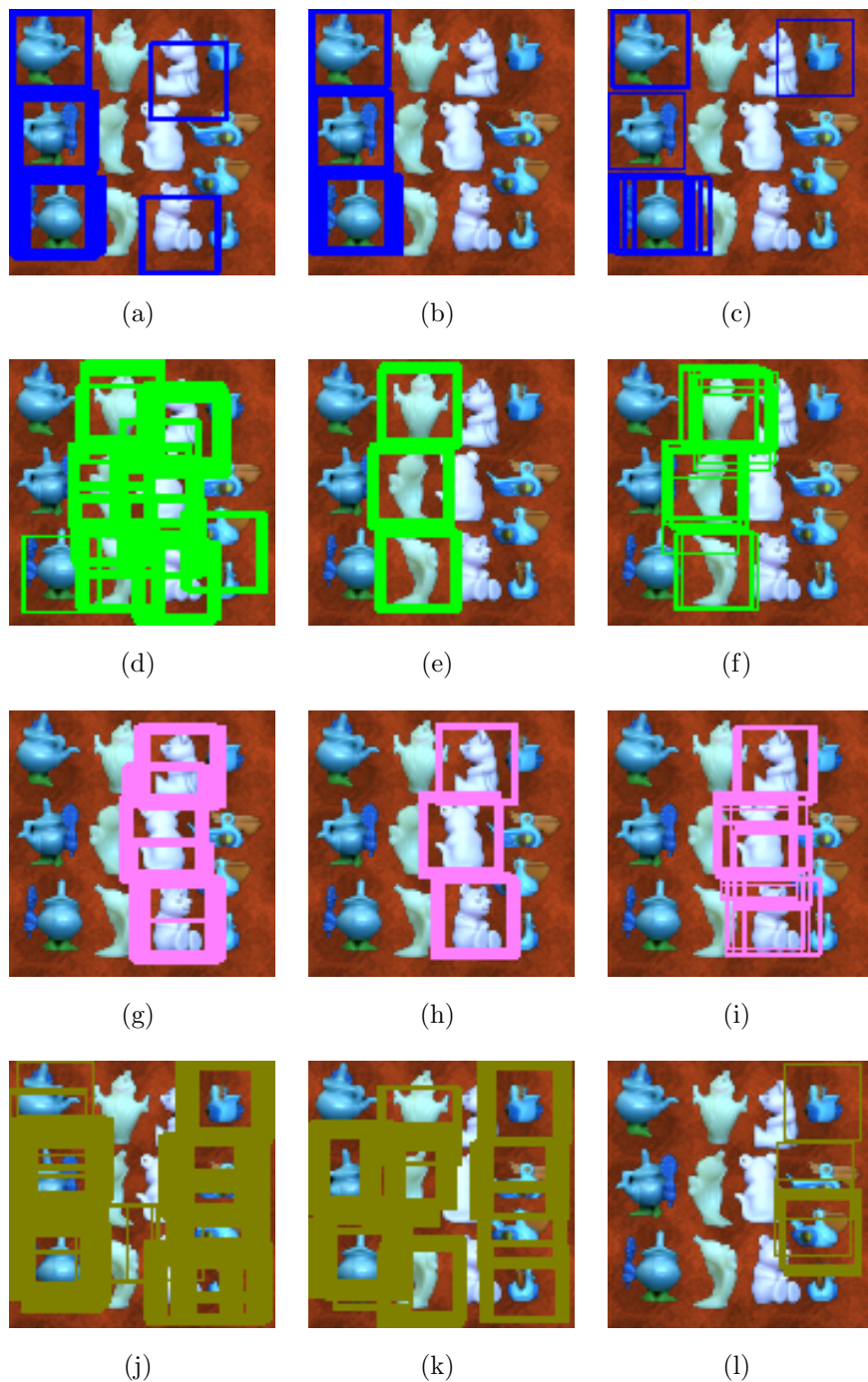
Tudi tu ugotovimo, da ima BW-metoda najvišjo stopnjo FP, sicer pa odkrije vsa mesta, kjer se nahajajo objekti. Metoda hierarhični SVM je najmanj občutljiva na spreminjanje ozadja, saj so rezultati praktično enaki za obe vrsti ozadja (7.15 in 7.16). Robustna PCA v primeru rdečkastega ozadja ne najde spodnjega pelikana (slika 7.16(1)). Z nižanjem praga glasovalne funkcije bi lahko dosegli, da bi metoda razpoznala pelikana tudi na tem mestu, vendar bi s tem znatno povečali število FP, kar ilustrira slika 7.14. Prag moramo določiti v skladu z nalogo (večji priklic ali večja natančnost), ki jo opravlja sistem za razpoznavanje.



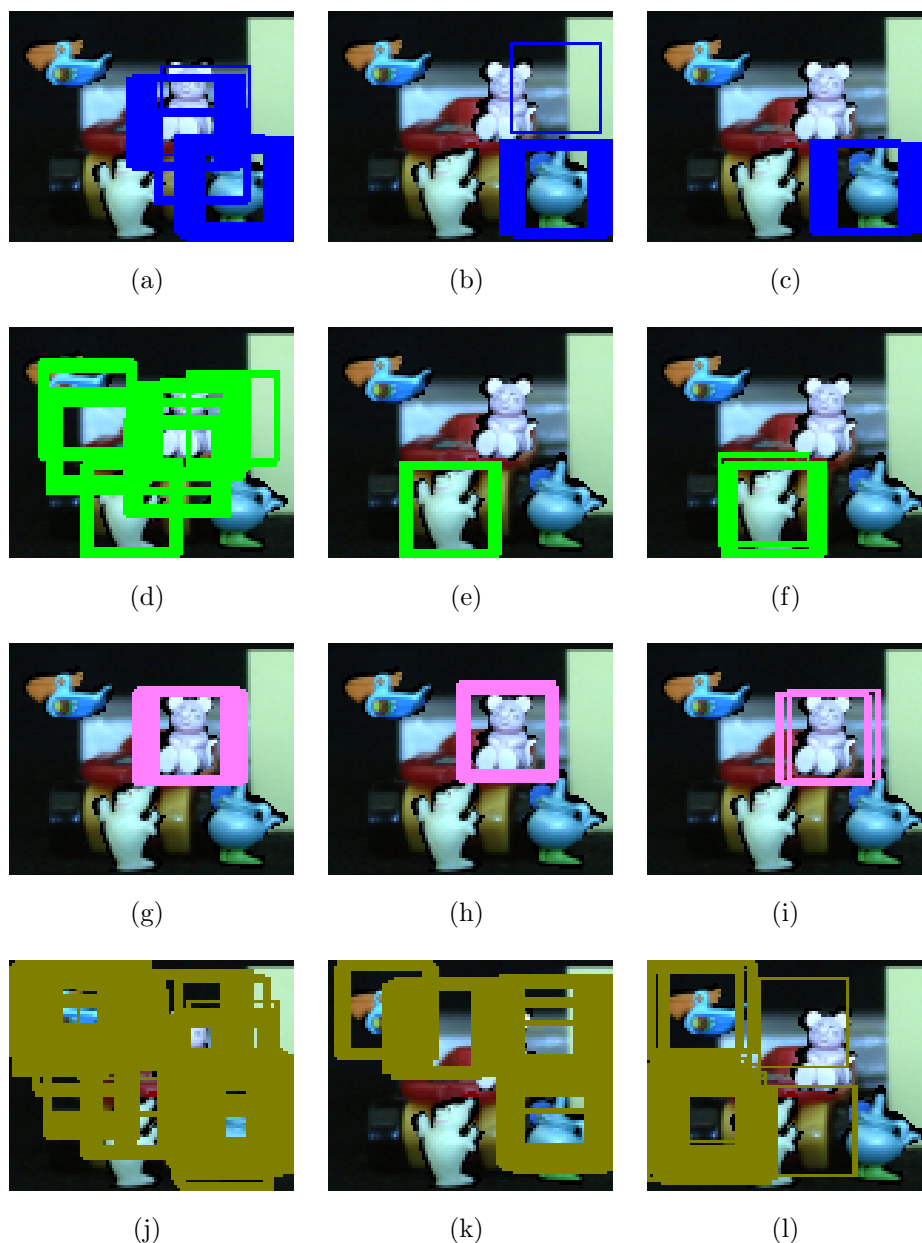
Slika 7.14: Rezultati za pelikana pri različnih pragih glasovalne funkcije. Uporabljeni pragovi od leve proti desni so: 0, 0.1, 0.3, 0.5 in 1.



Slika 7.15: Rezultati lokalizacije in klasifikacije na črnem ozadju. Prvi stolpec prikazuje rezultate BW-metode, drugi rezultate hierarhičnega SVM-ja in tretji rezultate robustnega PCA-ja. V prvi vrstici so z modrimi okviri označena mesta, kjer so metode na sliki razpoznale čajnik. V drugi vrstici z zelenimi okviri označujemo mesta, kjer so metode na sliki razpoznale duhca. Podobno z roza in oker okviri v tretji in četrti vrstici označujemo mesta, kjer so metode na sliki razpoznale miško oz. pelikana.



Slika 7.16: Rezultati lokalizacije in klasifikacije na rdečkastem ozadju. Prvi stolpec prikazuje rezultate BW-metode, drugi rezultate hierarhičnega SVM-ja in tretji rezultate robustnega PCA-ja. V prvi vrstici so z modrimi okviri označena mesta, kjer so metode na sliki razpoznale čajnik. V drugi vrstici z zelenimi okviri označujemo mesta, kjer so metode na sliki razpoznale duhca. Podobno z roza in oker okviri v tretji in četrti vrstici označujemo mesta, kjer so metode na sliki razpoznale miško oz. pelikana.



Slika 7.17: Rezultati lokalizacije in klasifikacije na realnem ozadju. Prvi stolpec prikazuje rezultate BW-metode, drugi rezultate hierarhičnega SVM-ja in tretji rezultate robustnega PCA-ja. V prvi vrstici so z modrimi okviri označena mesta, kjer so metode na sliki razpoznale čajnik. V drugi vrstici z zelenimi okviri označujemo mesta, kjer so metode na sliki razpoznale duhca. Podobno z roza in oker okviri v tretji in četrti vrstici označujemo mesta, kjer so metode na sliki razpoznale miško oz. pelikana.



Slika 7.19: 21 izbranih objektov iz COIL-100 baze



Slika 7.20: Primeri slik objekta obj 3 po normalizaciji velikosti in intenzitete

7.8 Barvna BW-, hierarhična SVM- in robustna PCA-metoda na COIL-100 bazi

Metode smo testirali na množici 21 objektov iz COIL-100-baze (slika 7.19). Za testiranje metod smo generirali dve testni sliki (slika 7.21), ki vključujeta 10 najključno izbranih slik različnih objektov iz testne množice. Testna slika 1 vsebuje objekte obj 1, obj 47, obj 24 in obj 20 (slika 7.22). Testna slika 2 vsebuje objekte obj 33, obj 3, obj 30, obj 2, obj 35 in obj 50 (slika 7.24).

Testirali smo tako, da smo okno velikosti 32×32 premikali po testni sliki in na vsaki lokaciji izvedli razpoznavo.



(a) Testna slika 1



(b) Testna slika 2

Slika 7.21: Testni sliki za objekte iz COIL-100 baze

7.8.1 Barvna BW-metoda

Učno množico slik smo povečali po principu barvne BW-metode. Na enak način kot pri testu na bazi figuric smo za vsako učno sliko generirali še eno z belim ozadjem, za vsako drugo učno sliko pa tudi slike z ekstremnimi RGB vrednostmi. Za razred ozadja smo uporabili iste slike kot pri testu na bazi figuric. Razred ozadja je torej vseboval 477 slik, razredi za objekte pa po 180 slik. Zgenerirali smo vse možne pare dvorazrednih SVM-jev za 22 razredov (21 objektov + razred ozadja), kar pomeni 531 dvorazrednih SVM-jev.

Rezultati barvne BW-metode na podmnožici objektov iz COIL-100 baze so grafično prikazani na slikah 7.23 in 7.25. Modra okna na slikah označujejo mesta, za katera je bilo okno razpoznano kot objekt, katerega ime se nahaja nad sliko.

Opazimo lahko, da so bili na testni sliki 1 vsi štirje objekti pravilno razpoznani. Tudi tu barvna BW-metoda generira veliko FP. Objekti obj 20, obj 22, obj 24, obj 33, obj 4, obj 41, obj 50, obj 51, obj 54, obj 64 in obj 65 so razpoznani na nepravilnih mestih. V večini primerov gre za barvno podobnost med iskanim objektom in področjem na sliki.

Na drugi testni sliki je bila barvna BW-metoda nuspešna pri razpoznavanju objektov obj 2 in obj 3. Ostale objekte je uspešno razpoznal (obj 30, obj 33, obj 35 in obj 50). Objekte obj 1, obj 22, obj 30, obj 33, obj 41, obj 51, obj 53, obj 64, obj 65 in obj 9 je razpoznala na mestih, kjer se niso nahajali.



Slika 7.22: Objekti na testni sliki 1



Slika 7.23: Grafični prikaz rezultatov za prvo testno sliko. Mesta, kjer barvna BW-metoda razpozna objekt, imenovan nad sliko, so označena z modrim okvirjem.



Slika 7.24: Objekti na testni sliki 2



Slika 7.25: Grafični prikaz rezultatov za drugo testno sliko. Mesta, kjer barvna BW-metoda razpozna objekt, imenovan nad sliko, so označena z modrim okvirjem.

7.8.2 Hierarhični SVM

Pri hierarhičnem SVM-ju postopamo na isti način kot pri testu z bazo figuric. Za vsak objekt posebej zgradimo hierahejo enorazrednih SVM-jev, za kritične pare pa še hierarhijo dvorazrednih SVM-jev. Rezultate za vsako testno sliko posebej prikazujeta sliki 7.26 in 7.27. Mesta, kjer hierarhična SVM razpozna objekt, imenovan nad sliko, so označena z modrim okvirjem,



Slika 7.26: Grafični prikaz rezultatov za prvo testno sliko. Mesta, kjer hierarhični SVM razpozna objekt, imenovan nad sliko, so označena z modrim okvirjem.

Opazimo, da hierarhični SVM deluje dobro tudi v primeru baze, ki vsebuje večje število

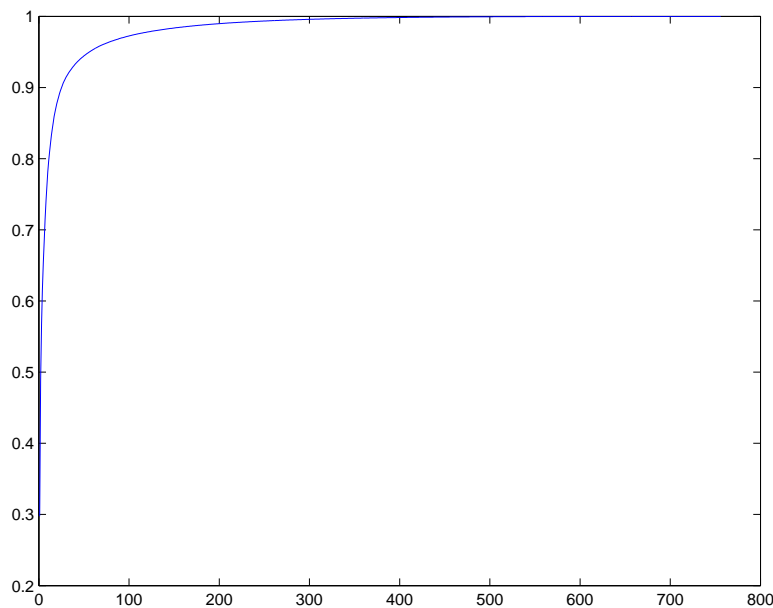


Slika 7.27: Grafični prikaz rezultatov za drugo testno sliko. Mesta, kjer hierarhični SVM razpozna objekt, imenovan nad sliko, so označena z modrim okvirjem.

objektov. Metoda je pravilno odkrila vse objekte. V primeru prve testne slike se je zmotila le enkrat, in sicer pri iskanju objekta obj 51, saj ga je razpoznala, čeprav ga na sliki sploh ni bilo. Do napake je najverjetneje prišlo zaradi barvne podobnosti med objektom 51 in področjem na sliki, kjer ga je metoda odkrila. Podobne rezultate smo dobili tudi na drugi testni sliki. Vsi objekti so pravilno razpoznani. Metoda se je zmotila le pri objektu obj 65. Tudi do te napake je prišlo zaradi podobnosti objekta 65 s področjem na sliki, kjer ga je metoda napačno razpoznala.

7.8.3 Robustna PCA-metoda

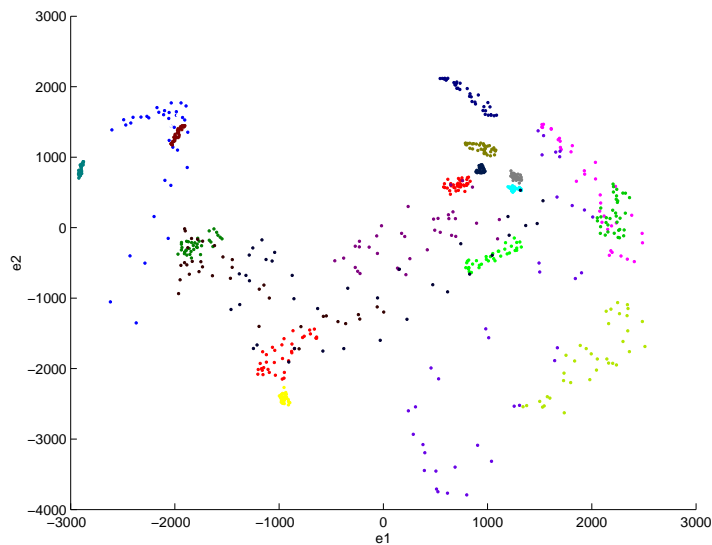
Učenje in razpoznavanje s pomočjo robustne PCA-metode je potekalo tudi pri poskusih na COIL-100 bazi podobno kot na bazi fuguric. Izračunali smo lastni prostor. Pri preslikovanju slik v lastni prostor smo upoštevali 15 lastnih vektorjev, kar pomeni 84% energije sistema (slika 7.28). Projekcije slik objektov iz COIL-100 baze v prostor, ki ga razpenjata prva dva oz. prvi trije lastni vektorji so prikazane na slikah 7.29 in 7.30.



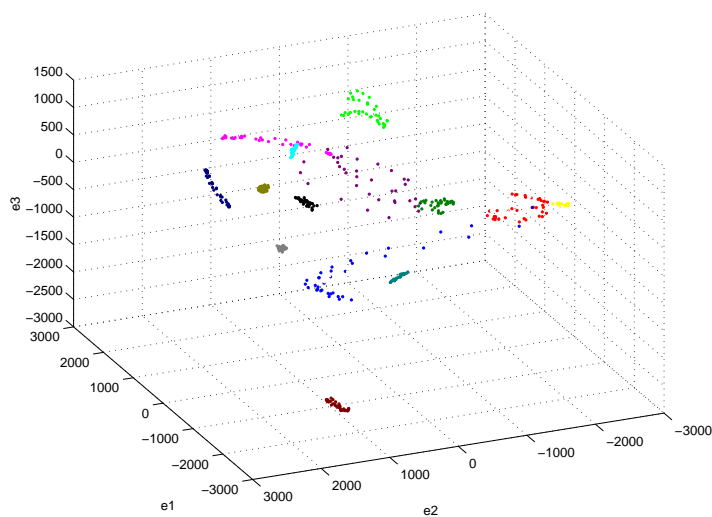
Slika 7.28: Število lastnih vektorjev in pripadajoče energije v primeru COIL-100 baze

Maske, ki smo jih uporabljali pri usmerjanju izbire slikovnih elementov, so predstavljale preseke in unije vseh 21 predmetov (slika 7.31). Za vsako sliko smo torej izračunali 42 hipotez. Pri generiranju hipotez smo začeli s 450 točkami in jih na koncu ohranili vsaj 300 ($\alpha = 0,9$). Več točk smo vzeli iz dveh razlogov. Prvi je, da smo računali z večjim številom lastnih vektorjev in smo tako pri nastavitvi enačb potrebovali več točk. Drugi razlog pa izhaja iz lastnosti COIL-100 baze, kjer projekcije predmetov zavzamejo več slikovnih elementov na sliki kot v primeru baze figuric.

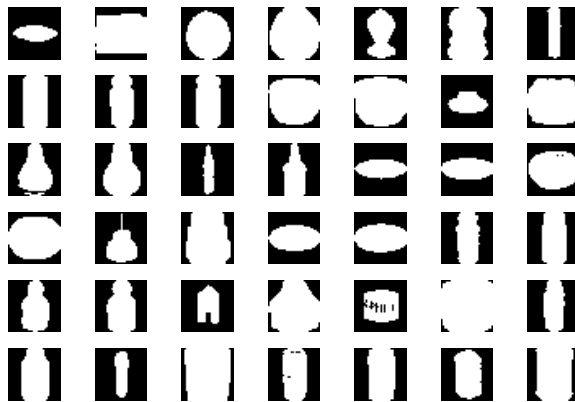
Rezultati za robustno PCA-metodo so prikazani na slikah od 7.32 do 7.35. Na slikah 7.32(a) in 7.33(a) so prikazane vse hipoteze, ki so preživele. Vsaki lokaciji na sliki ustreza



Slika 7.29: Projekcije slik objektov iz COIL-100-baze v podprostor, določen s prvima dvema lastnima vektorjema.



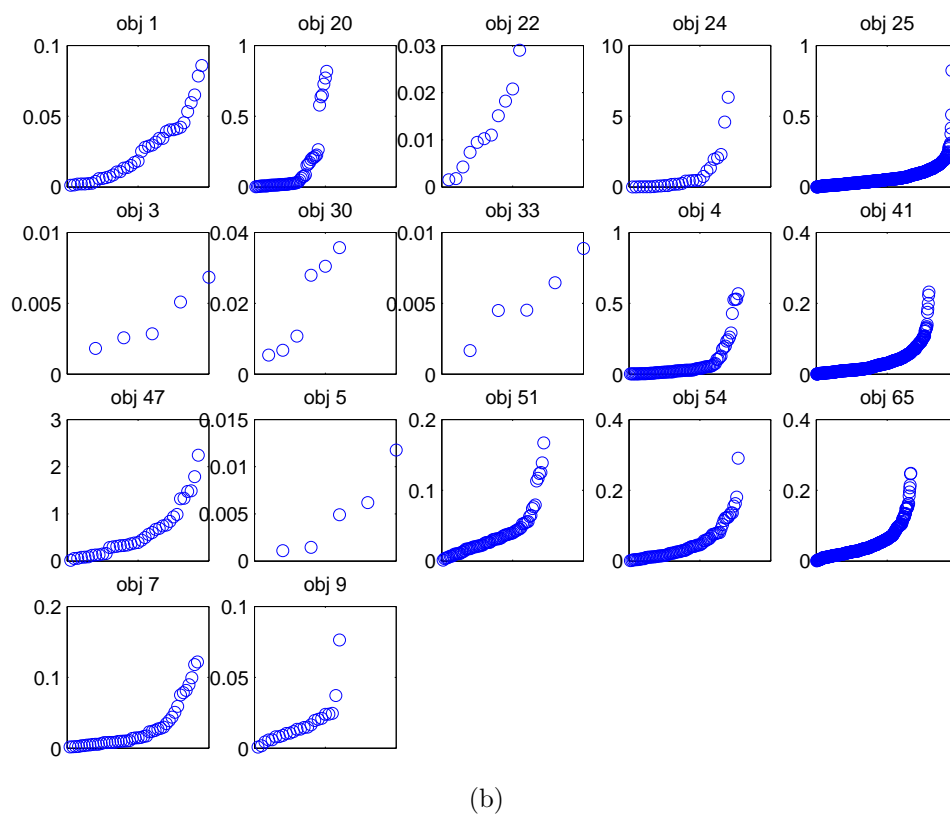
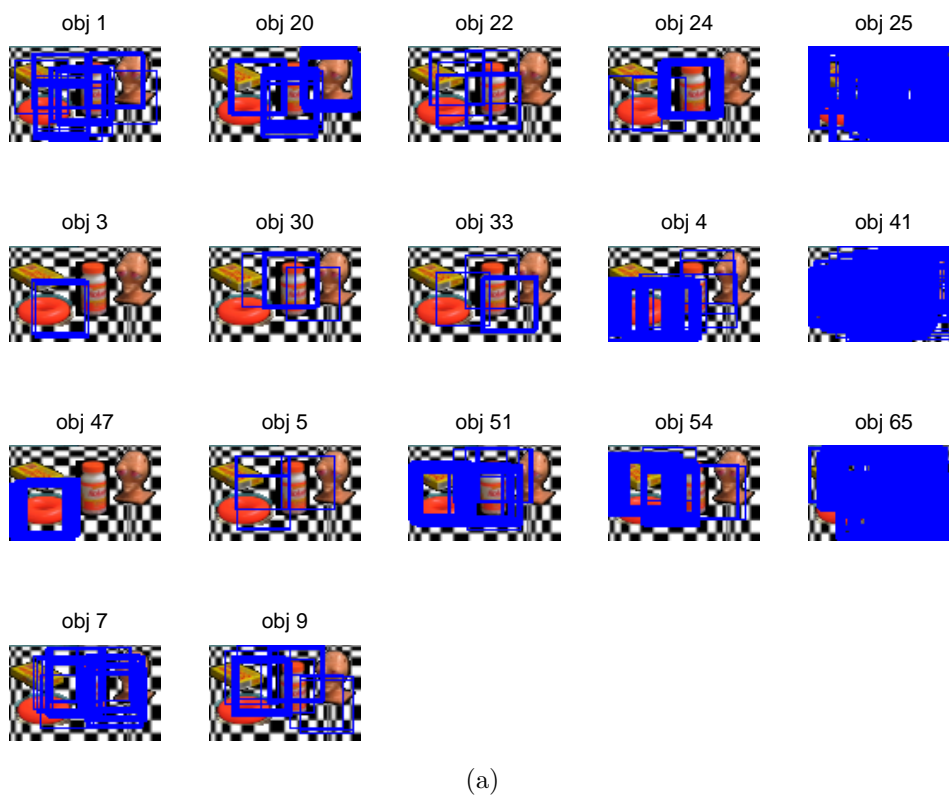
Slika 7.30: Projekcije slik objektov iz COIL-100-baze v podprostor, določen s prvimi tremi lastnimi vektorji.



Slika 7.31: Uporabljene maske v primeru COIL-100 baze

eno modro okno. Modro okno je dodeljen objektu, ki mu je glasovalna funkcija dodelila največ glasov. Sliki 7.32(b) in 7.33(b) za vsako testno sliko posebej prikazujeta vrednosti glasovalne funkcije za vsak objekt posebej. Ker pri robustni PCA nimamo razreda ozadja, metoda na vseh mestih razpozna enega izmed objektov, vendar pa so mnogi od teh v resnici ozadje. Opazimo lahko, da imajo objekti, ki jih v resnici ni na sliki, manjše vrednosti glasovalne funkcije. To lastnost izkoristimo tako, da vpeljemo prag, ki določa, kdaj je objekt na sliki v resnici odkrit in kdaj ne. S pragom uravnotežimo razmerje med FP in TP.

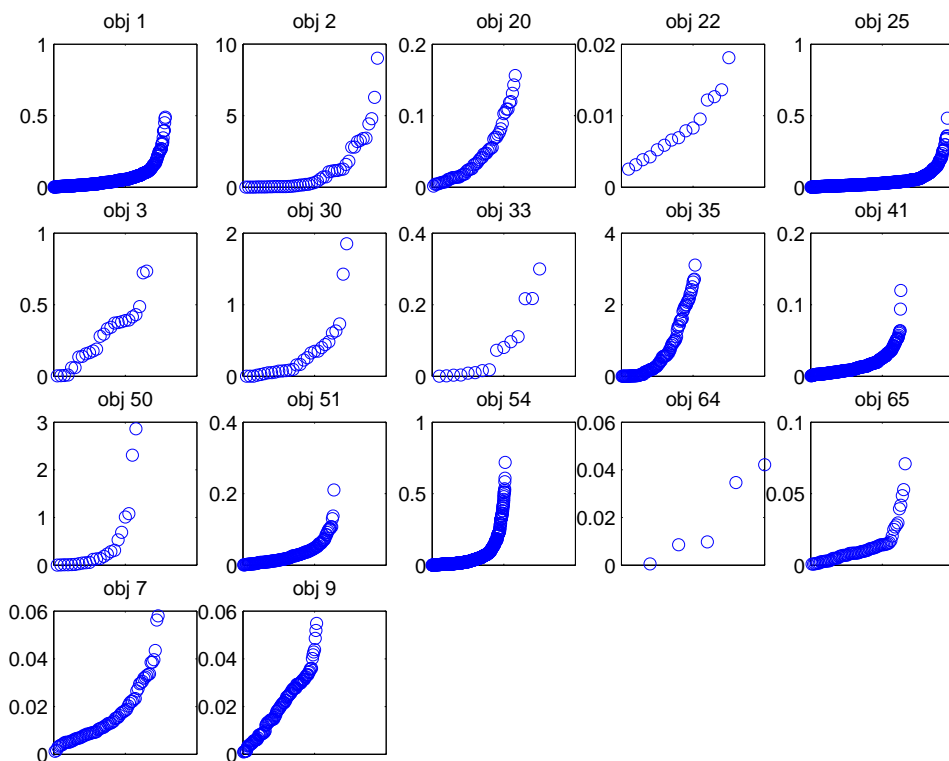
Sliki 7.34 in 7.35 prikazujeta rezultate, če za prag vzamemo vrednost 0,5. Metoda uspe na testni sliki 1 pravilno razpoznati tri objekte. Objekt obj 1 ostane nerazpoznan. Metoda zaradi barvne podobnosti napačno razpozna obj 25 in obj 4. Na testni sliki 2 uspe razpoznati 5 objektov. Manjkajoči objekt obj 33 je uvrščen med kandidate, vendar zaradi nizkih vrednosti glasovalne funkcije ni potrjen. Objekt obj 54 je zaradi barvne podobnosti s področjem na testni sliki razpoznan na področju, kjer ga ni.



Slika 7.32: Rezultati robustne PCA-metode na testni sliki 1. (a) Mesta, ki jih robustna PCA-metoda dodeli objektu, imenovanemu nad sliko, so označena z modrim okvirjem. (b) Vrednosti glasovalne funkcije so za vsak objekt posebej urejene v naraščajočem vrstnem redu. Vsak krogec v grafu pripada enemu oknu na ustrezni zgornji sliki.



(a)



(b)

Slika 7.33: Rezultati robustne PCA-metode na testni sliki 2. (a) Mesta, ki jih robustna PCA-metoda dodeli objektu, imenovanemu nad sliko, so označena z modrim okvirjem. (b) Vrednosti glasovalne funkcije so za vsak objekt posebej urejene v naraščajočem vrstnem redu. Vsak krogec v grafu pripada enemu oknu na ustrezni zgornji sliki.



Slika 7.34: Rezultati robustne PCA-metode na testni sliki 1. Mesta, ki jih robustna PCA-metoda dodeli objektu, imenovanemu nad sliko, so označena z modrim okvirjem. Prag za glasovalno funkcijo je 0,5.



Slika 7.35: Rezultati robustne PCA-metode na testni sliki 1. Mesta, ki jih robustna PCA-metoda dodeli objektu, imenovanemu nad sliko, so označena z modrim okvirjem. Prag za glasovalno funkcijo je 0,5.

Poglavje 8

Zaključek in nadaljnje delo

V delu rešujemo problem razpoznav 3-D objektov na poljubnem ozadju in pri različnem zornem kotu zajemanja slike. Omejili smo se na glediščno osnovane metode. Razvili smo hierarhično SVM-metodo in jo primerjali z barvno BW- in robustno PCA-metodo. V procesu razpoznav so nepoznani zorni kot, objekt in ozadje. Glavni problem, ki pri razpoznavanju objektov na slikah nastopi, je, da ne vemo vnaprej, kateri slikovni elementi pripadajo ozadju in kateri objektu. Gre za problem segmentacije regij, ki pripadajo objektu, ki ni vnaprej določen.

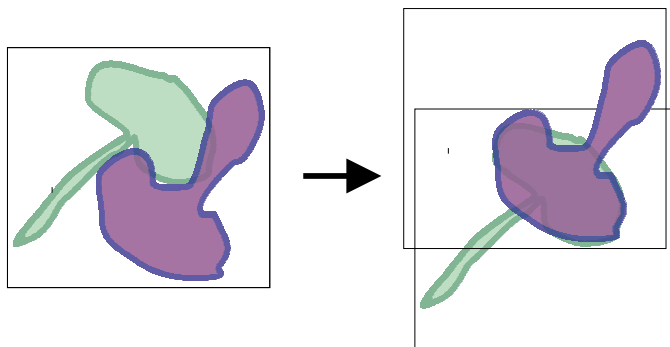
V nasprotju s fazo razpoznav pa lahko v učni fazi nadzorujemo pogoje pod katerimi zajemamo slike. V našem primeru so se objekti nahajali na črnem ozadju, kar je omogočalo enostavno segmentacijo v fazi učenja. To smo pri različnih metodah izkoriščali na različne načine. Pri barvni BW-metodi smo razširili učno množico z menjavo ozadja, pri hierarhičnem SVM-ju in robustnem PCA-ju pa smo na podlagi segmentiranih učnih slik v fazi razpoznavanja usmerjali izbiro podmnožice slikovnih elementov na testni sliki, na kateri je potem temeljila klasifikacija.

Tri preizkušene metode temeljijo na zelo različnih principih. Barvna BW-metoda uporablja pristop pedagoškega učenja, kjer so primeri v učni množici pripravljeni tako, da zajamejo slikovni elementi ozadja maksimalno variabilnost. Hierarhična metoda sloni za strukturiranjem zaporedju večjega števila klasifikatorjev. PCA pa sloni na večjem številu hipotez, kjer s pomočjo mask usmerimo izbiro podatkov na različne dele slike in upamo, da bo vsaj ena hipoteza zajela večino podatkov, ki pripadajo objektu.

V primeru hierarhičnega SVM-ja se pomikamo po hierarhiji navzdol. Pri tem večamo število slikovnih elementov, ki jih vključujejo klasifikatorji, in hkrati manjšamo število slik, ki bi lahko ustrezale posameznemu klasifikatorju, saj zahtevamo skladnost na večjem področju slike. Da na sliki najdemo predmet, morajo zanj glasovati vsi klasifikatorji vsaj na eni poti od korena do lista hierarhične strukture. Poleg razpoznave objekta na sliki omogoča ta princip tudi določitev okvirne orientacije objekta na sliki, če sledimo veji v hierarhiji enorazrednih SVM-jev, ki je potrdila razpoznavo objekta.

Hierarhični SVM bi lahko še dodatno izboljšali na več načinov. Eden izmed njih je, da stopnjo hierarhične strukture prilagodimo značilnostim posameznega objekta. Enostavni objekti, ko je npr. enobarvna žoga, ne potrebujejo hierarhične predstavitve. Dovolj je torej en nivo. Kompleksni predmeti, ki so zelo razčlenjenih oblik in pisanih tekstur pa verjetno v nekaterih primerih potrebujejo več kot trinivojsko hierarhijo.

Za hierarhični SVM je dobro, če se področja na sliki, ki predstavljajo objekte, čim bolj prekrivajo. Iz tega vidika bi bilo smiselno, da bi objekte po učnih slikah premikali tako, da bi dobili največje preseke med področji, ki predstavljajo objekte na slikah (slika 8.1). Seveda se pri takem načinu pojavijo nekatera dodatna vprašanja, kot je npr., kako obravnavati dele objekta, ki pri takem premikaju učnih slik padejo iz slike itd.



Slika 8.1: Učne slike poravnamo tako, da objekti med seboj tvorijo največje možne preseke.

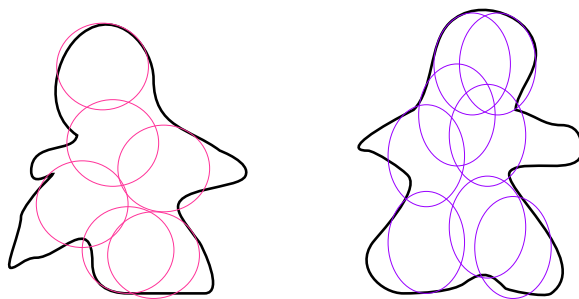
Pri BW-metodi, ki smo jo realizirali s pomočjo teniškega turnirja, čas gradnje vseh potrebnih SVM-jev narašča eksponentno. Če bi sistem zahteval ločevanje med velikim številom objektov, bi bilo smiselno izbrati drugi pristop k reševanju večrazrednega problema, npr. pristop eden proti vsem ostalim.

Če metode analiziramo s stališča lokalnosti in globalnosti, ugotovimo, da bi BW-metodo uvrstili med globalne pristope, saj tako v fazi učenja kot v fazi razpoznavanja potrebuje slikovne elemente celotne slike. Ostali dve metodi, hierarhični SVM in robustni PCA, pa imata tudi lokalni pridih. Pri hierarhičnem SVM-ju potekata faza učenja in razpoznavanja na različnih podmnožicah slikovnih elementov. Podmnožice so fiksne in jih določimo v fazi učenja. Robustni PCA za predstavitev objektov sicer potrebuje celotne učne slike, vendar pa rekonstrukcija in razpoznavanje slonita le na podmnožici slikovnih elementov. Podmnožice tu niso fiksno določene. Smiselno je, da jih določimo tako, da izberemo le tiste dele, ki ne vsebujejo šuma. V našem primeru je to pomenilo izbiro slikovnih elementov, ki pripadajo objektu. Podmnožice smo v postopku robustnega PCA izbirali na podlagi mask, ki določajo potencialna področja, iz katerih izbiramo podmnožice slikovnih elementov. Vendar pa tudi v primeru robustnega PCA-ja iščemo globalne koeficiente, ki sliko predstavljajo kot celoto.

Smiselno je, da objekt predstavimo ne le kot celoto, ampak tudi s pomočjo lokalnih značilnosti. Seveda pa je izbira dobrih — diskriminativnih, lokalnih značilnosti precej težka naloga. V našem primeru, ko smo rokovali s slikami velikosti 32×32 , bi lokalnost težko uporabili, saj so slike za to premajhne. Detajlne lastnosti na slikah z nizko resolucijo so zabrisane. Pri večji resoluciji bi bili tudi posamezni objekti med seboj bolj različni. Posledica bi bila večja klasifikacijska točnost in manjša stopnja FP. Možnosti, kako v predstavitev vkomponirati lokalnost, je več.

Lepa lastnost vseh treh sistemov je, da jih lahko enostavno nadgradimo, če dodajamo nove objekte. Pri barvni BW-metodi izračunamo in dodamo ustrezne SVM-je, pri hierarhičnem pristopu dodamo nove hierarhije, pri robustnem PCA pa uporabimo inkrementalno metodo gradnje lastnega prostora [23]. Vendar pa s številom objektov v bazi narašča tudi kompleksnost problema, zato bi bilo smiselno nekako predmete grupirati v skupine in najprej izvajati razpoznavo med skupinami in šele nato razpoznavanje v skupini sami. Tak pristop bi bil najbolj smiselni pri hierarhičnem pristopu, saj ta sedaj zahteva iskanje vsakega predmeta s svojo hierarhijo. S predlaganim pristopom pa bi najprej iskali predmete s hierarhijami skupin, ki bi jih bilo manj, kot je vseh objektov. Ko bi enkrat določili, v katero skupino spada objekt na testni sliki, bi nadaljevali razpoznavanje s hierarhijami posameznih objektov, ki spadajo znotraj izbrane skupine.

V nadaljnjih raziskavah načrtujemo učenje SVM-jev na poldagi lokalnih delov posameznega objekta. Osnovna ideja je, da bi kot učno množico generirali vse možne podmnožice slikovnih elementov določene oblike — maske, kot to nakazuje slika 8.2. Če bi slikovni elementi, ki bi jih zajela maska, delno pripadali tudi ozadju, bi se lahko poslužili pedagoškega učenja, kot ga predlaga BW-metoda. Izmed vseh maskiranih delov slik bi morali izločiti tiste, ki so si zelo podobni, a predstavljajo različne predmete. V fazi testiranja bi na testnih slikah iskali mesta, kjer bi dobili dovolj pozitivnih odzivov za nek objekt. Za taka mesta bi lahko sklepali, da se tam nahaja poznan objekt.



Slika 8.2: Maske, ki določajo lokalne značilke predmeta

Aplikacij, ki zahtevajo razpoznavanje objektov v različnih situacijah, je kar nekaj: razpoznavanje prometne signalizacije pri avtonomni vožnji, analiza slik kromosomov, avtomatsko sortiranje izdelkov, sledenje posameznemu igralcu na igrišču itd. Metode, ki bi obvladale vse možne situacije, še ni. Smiselno je, da analiziramo specifičnosti posameznih problemov in temu primerno prilagodimo metode. Za različne vrste aplikacij so primerne različne metode.

Poglavje 9

Dodatek A

9.1 Notacija

- Skalarje pišemo poševno tiskano (npr. b)
- Vektorje pišemo z malimi, odebeljenimi črkami (npr. \mathbf{w} , $\boldsymbol{\alpha}$)
- Matrike pišemo z velikimi črkami in krepko odebeljeno pisavo (npr. \mathbf{X})
- N - število učnih slik
- M - število slikovnih elementov v sliki oz. dimenzija vhodnega prostora; $M = r \times c \times d$
- l - število slikovnih elementov v eni vrstici slike
- c - število stolpcev v sliki
- d - število elementov, ki v matriki predstavljajo en slikovni element. Pri sivinskih slikah ima d vrednost 1, pri RGB slikah pa vrednost 3.
- M' - dimenzija prostora značilk
- s - število podpornih vektorjev (SV)

- \mathcal{X} - vhodni prostor
- \mathcal{H} - prostor značilk
- Φ - preslikava iz vhodnega prostora v prostor značilk
- \mathbf{x} - vektorji vhodnega prostora
- \mathbf{z} - vektorji iz prostora značilk
- M_{bg} - število slikovnih elementov, ki predstavljajo ozadje
- K - maska
- m - stopnja podobnosti
- p - zamik učne množice pri hierarhičnih dvorazrednih SVM-jih

Zahvala

Iskreno se zahvaljujem svoji mentorici, docentki dr. Jasni Maver, za vse ure, ki jih je presedela z mano za računalnikom, za njeno nalezljivo zagnanost pri raziskovalnem delu in za koristne napotke.

Zahvaljujem se prof. dr. Alešu Leonardisu, ki s svojim širokim znanjem, z izkušnjami in delovno vnemo največ pripomore h kvalitetnemu delu.

Verjamem, da delo v prijateljskem ozračju pozitivno vpliva na delovne uspehe. Prav gotovo je LRV eno najprijetnejših delovnih mest. Zahvaljujem se Bojanu za nasvete pri pisanju v \LaTeX -u, Mateju, Danijelu, Matjažu, Borutu, Mihi in ostalim za dobro voljo in pomoč v vsakem trenutku.

Zahvala velja tudi članom CMP iz Prage, ki so se izkazali kot prijetni gostitelji. Na tem mestu se še posebej zahvaljujem Vojtěchu Francu za napotke v zvezi s SVM-ji in dr. Jiriju Matasu za vse kritične pripombe.

Na koncu se zahvaljujem najbližjim, ki so verjeli, da bom začeto delo izpeljala do konca in se z mano veselijo mojih uspehov.

Literatura

- [1] H. Bischof, H. Wildenauer, and A. Leonardis. Illumination insensitive eigenspaces. In *Proc. ICCV01*, pages 233–238. IEEE Computer Society, 2001.
- [2] C. J. C. Burges. Simplified support vector decision rules. In *International Conference on Machine Learning*, pages 71–77, 1996.
- [3] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [4] Y. Chen, X. Zhou, and T. Huang. One-class SVM for learning in image retrieval. In *Proceedings of the 2001 IEEE International Conference On Image Processing (ICIP-01)*, pages 34–37, Thessaloniki, Greece, October 7–10 2001. IEEE.
- [5] V. Cherkassky and F. Mulier. *Learning from Data : Concepts, Theory, and Methods*. Wiley-Interscience, 1998.
- [6] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, second edition, October 2000.
- [7] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 47–57, 1984.
- [8] C. Hsu and C. Lin. A comparison of methods for multi-class support vector machines. Technical report 19, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001.

- [9] M. J. Jones and P. Viola. Robust real-time object detection. Technical Report CRL-2001-1, Hewlett Packard Laboratories, February 25 2001.
- [10] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding: CVIU*, 78(1):99–118, April 2000.
- [11] A. Leonardis, A. Gupta, and R. Bajcsy. Segmentation of range images as the search for geometric parametric models. *International Journal of Computer Vision*, 14(3):253–277, April 1995.
- [12] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In P. L. Rosin and D. Marshall, editors, *Proceedings of the British Machine Vision Conference*, pages 384–393, London, UK, September 2002. BMVA. volume 1.
- [13] S. Nayar, H. Murase, and S. Nene. Parametric appearance representation. In *Early Visual Learning*, pages 131–160. Oxford University Press, 1996.
- [14] S. Nene, S. Nayar, and H. Murase. Columbia object image library: Coil. Technical Report CUCS-006-96, Department of Computer Science, Columbia University, February 1996.
- [15] Š. Obdržálek and J. Matas. Object recognition using local affine frames on distinguished regions. In P. L. Rosin and D. Marshall, editors, *Proceedings of the British Machine Vision Conference*, pages 113–122, London, UK, September 2002. BMVA. volume 1.
- [16] C. Papageorgiou, T. Evgeniou, and T. Poggio. A trainable pedestrian detection system. In *Intelligent Vehicles*, pages 241–246, October 1998.
- [17] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin dags for multiclass classification. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 547–553, 2000.

-
- [18] D. Roobaert. *Pedagogical Support Vector Learning: a pure learning approach to object recognition*. PhD thesis, Royal Institute of Technology (KTH), Dept. of Numerical Analysis and Computing Science (NADA), Stockholm, Sweden, 2001.
 - [19] B. Schölkopf and J. Shawe-Taylor Robert C. Williamson, A. Smola. Sv estimation of a distribution's support. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Neural Information Processing Systems 12*. MIT Press, 1999.
 - [20] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002.
 - [21] A. Selinger and R. C. Nelson. A perceptual grouping hierarchy for appearance-based 3D object recognition. *Computer Vision and Image Understanding: CVIU*, 76(1):83–92, 1999.
 - [22] A. Selinger and R. C. Nelson. Improving appearance-based object recognition in cluttered backgrounds. Technical Report TR725, University of Rochester, 2000.
 - [23] D. Skočaj. *Robust Subspace Approaches to Visual Learning and Recognition*. PhD thesis, Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenija, 2002.
 - [24] J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.

Izjava

Izjavljam, da sem magistrsko delo izdelala samostojno pod vodstvom mentorice doc. dr. Jasne Maver. Izkazano pomoč drugih sodelavcev sem v celoti navedla v zahvali.